REDUCED ORDER APPROXIMATIONS TO HIGHER ORDER LINEAR SYSTEMS

Jerry Dennis Thompson

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

REDUCED ORDER APPROXIMATIONS
TO
HIGHER ORDER LINEAR SYSTEMS

by

Jerry Dennis Thompson

Thesis Advisor: A. Gerba Jr.

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) REDUCED ORDER APPROXIMATIONS TO HIGHER ORDER LINEAR SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED Masters Thesis; June 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerry Dennis Thompson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 79
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Reduced Linear Approximations
Approximation methods
Reduced Equations
Computer Approximations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Low order models are derived by a computer program technique which utilizes the Routh Approximation Method of analysis. Comparisons are made between this method and that of the Dominant Pole Method and the Iterative Optimization Method of analysis.

Low order models are developed from higher-order

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601 |

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

linear systems and compared to that system in response to input excitations consisting of a Step and a Ramp.

Graphical displays and numerical tables provide a basis for error analysis and comparisons between the approximation techniques.

REDUCED ORDER APPROXIMATIONS TO HIGHER ORDER LINEAR SYSTEMS

by

ennis
Jerry D. Thompson
Lieutenant, United States Navy
B.S., University of New Mexico, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

June 1976

ABSTRACT

Low order models are derived by a computer program technique which utilizes the Routh Approximation Method of analysis. Comparisons are made between this method and that of the Dominant Pole Method and the Iterative Optimization Method of analysis.

Low order models are developed from higher-order, linear systems and compared to that system in response to input excitations consisting of a Step and a Ramp.

Graphical displays and numerical tables provide a basis for error analysis and comparisons between the approximation techniques.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

TABLE OF SYMBOLS

Symbol	Definition
A	Original seventh-order equation
R	Routh approximation
D	Dominant Pole approximation
O	Optimum minimization equation
M_{pt}	Peak overshoot = Y_{max}/Y_{ss}
T_d	Delay time
T_r	Rise time
T_s	Settling time
E	Error=A-Approximant (response)
J	Average absolute error
C_d	Denominator coefficients (Routh Table)
C_n	Numerator coefficients (Routh Table)

ACKNOWLEDGEMENT

The author wishes to express sincere appreciation and gratitude to Professors A. Gerba Jr. and George J. Thaler of the Electrical Engineering Department, Naval Postgraduate School of Monterey, California, for their advice and guidance.

I. INTRODUCTION

The complexity involved in designing a control system of reasonably low order for dynamic systems of substantially higher order is a subject about which much has been written. The list of references at the end of this study include only a small portion of the material available concerning this subject. A new development has been introduced by Maurice F. Hutton and Bernard Friedland, [Ref 1], which provides a systematic and analytical approach to obtaining reduced-order approximations from higher-order transfer functions. This method is referred to as the "Routh Approximation Method," and is based on an expansion that uses the Routh table of the original higher-order transfer function. The desire for low order models to simulate higher-order practical systems, such as electrical power plants, chemical processes, aircraft model designs and electronic circuitry is obvious when dealing with such complex systems. The "Routh Approximation Method", is thus, an extremely powerful tool to utilize in dealing with these types of problems.

In comparison to other available methods presently used, the Routh method has many advantages. The widely used "Dominant Pole", approximation method , which is based on approximating a system by utilizing the poles, or characteristic roots nearest the imaginary axis, has the disadvantage that the roots of the characteristic equation must be found. For a very-high-order system, this is not a trivial task. The "Pad'e" approximation is based on setting the numerator and denominator orders to a desired value, the coefficients are then chosen so that the Taylor series expansions of the approximant and the original transfer

function agree in as many terms as possible. This method produces accurate results, however, it is limited in application to single-input, single-output system analysis. In addition, an unstable approximant may be obtained from a stable system since the approximant's poles depend on both the original equation's numerator and denominator.

The "Routh Approximation Method," preserves stability if the original transfer function is stable. It provides an efficient means of obtaining lower-order approximants for multiple inputs or outputs and is very adaptable to computer programming.

This thesis is concerned with the development of a computer program which utilizes the "Routh Approximation Method," to obtain lower-order transfer functions from higher-order functions and compares the resulting equations to the original equations by graphically displaying the response of each to various input excitations. The program was designed to provide output data which is useful in determining which degree of approximant is best suited to simulate the original higher-order equation. The orders available are the first through the fourth order reduced transfer functions. The program is capable of reducing transfer functions up to, and including the tenth order and, without loss of generality, may be extended to handle an almost unlimited order.

For illustrative purposes, a seventh order system was used, as an example, to demonstrate the computer program's capabilities. This particular system was chosen merely to demonstrate the simplicity involved in utilizing the program and has no physical relationship to any real system.

II. NATURE OF THE PROBLEM

In approximating higher-order systems by lower-order models, a linear system is desirable since the linear characteristic equations are less complicated than the non-linear equations. Therefore, this study is primarily concerned with linear, time invariant characteristic equations, since the objective in developing the computer program was to illustrate a simple analytical approach to obtaining lower order characteristic equations from higher order systems.

It is obvious that a reduced-order model cannot characterize a given system as accurately as a higher-order model. The validity of the lower-order model is based upon its degree of success in approximating the higher order system in representing the characteristics of primary interest.

Interpreting the solution of a higher-order system often results in computational difficulties which are reduced by appropriate selection of a reduced-order approximation.

Ideally, a reduced-order model would approximate the higher-order system in both low and high frequency ranges. In doing so, some accuracy is lost in compensating for the different responses of the system to variations in frequency.

The low frequency model more closely approximates the higher-order system than a model composed of both low and high order frequency characteristics. The procedure

utilized in this study places emphasis on the low frequency model.

For unstable systems, the program described herin is still valid, however, the original higher-order system must first be modified by shifting the imaginary axis prior to the approximation. Thus for an unstable transfer function, H(s), the equation must be changed to H(s+a), where a is chosen sufficiently large so that H(s) is asymptotically stable. This procedure is described in detail in [Ref 1], pp 332.

III. COMPUTER PROGRAM CRITERIA

A. GENERAL

The program, referred to as ROUTH1, was written with the following criteria:

1. Minimum utilization of computer time

ROUTH1 consists of less than 150K of storage and takes less than 2 minutes of computer time. This not only conserves efficiency, but also provides the user with the desired data at a minimum cost.

2. Ease of use

Input data required consists of eleven data cards for maximum utilization of the program. Emphasis is placed on the ease of using the program to obtain the desired results with minimum time expended on computer programing.

3. Usefullness of output

a. The first, second, third and fourth order approximants to the original equation are printed in transfer function format. Both numerator and denominator coefficients are printed in ascending powers of S.

b. The roots of the original and reduced equations are provided to enable the user to study the response of the systems in the frequency domain.

c. Choice of variables for print out in table form is available with up to eight variables maximum for any one of three possible runs.

d. Choice of variables for graphical output with up to four curves may be plotted separately or all on one graph.

e. Graphical output response to input excitations consisting of a Step, Ramp, or Sinusoidal input are available. The graphs display the original output response compared to the lower order response and the error is plotted to display the differences in response. Multiple inputs may be used.

B. SUBROUTINES

Two subroutines are utilized in ROUTH1. The roots of the original and reduced equations are determined by subroutine PRQD , which was taken from [Ref 3], and the tables and graphs are determined by subroutine REDUCT1 which is a modification of INTEG1 , [Ref 3].

IV. FOURTH ORDER EXAMPLE

A. THREE STEP PROCEDURE

To illustrate the "Routh Approximation Method", A fourth order example was chosen for simplicity.

The method, described in detail in [Ref 1], consists of three basic steps. The following transfer function is used to illustrate the procedure:

$$H(s) = \frac{1}{20 + 32S + 24S^2 + 8S^3 + S^4}$$

The first step is to compute what are termed Alpha and Beta coefficients from the Routh Table shown on page 19.

	20	24	1
Alpha	32	8	
$Alpha_1$=0.625	19	1	
$Alpha_2$=1.684	6.3158		
$Alpha_3$=3.008	1		
$Alpha_4$=6.3158			

Beta	1	0
	0	0
$Beta_1$=0.03125	-0.25	
$Beta_2$=0.0	0.0	
$Beta_3$=0.03958		
$Beta_4$=0.0		

Step two in the procedure is to obtain what are termed the Routh convergents, which are based on the following:

Letting $A_k(s)$ and $B_k(s)$ denote the denominator and the numerator, respectively, of the k^{th} Routh convergent, i.e.,

$$A_1(s) = Alpha_1 S + 1$$

$$B_1(s) = Beta_1$$

$$A_2(s) = Alpha_1 Alpha_2 S^2 + Alpha_2 S + 1$$

$$B_2(s) = Alpha_2 Beta_1 S + Beta_2$$

$$\vdots$$

The general expression from [Ref 1] is the following:

$$A_k(s) = Alpha_k(s) A_{k-1}(s) + A_{k-2}(s)$$

$$B_k(s) = Alpha_k S B_{k-1}(s) + B_{k-2}(s) + Beta_k \qquad k=1,2,3...$$

$$\text{with} \quad A_{-1}(s)=0.0 \qquad B_{-1}(s)=0.0$$

$$A_0(s) = 1.0 \qquad B_0(s) = 0.0$$

The Routh convergents for the fourth order example are the following:

$$R_1(s) = \frac{0.03125}{0.625S + 1}$$

$$R_2(s) = \frac{0.056316}{1.05263S^2 + 1.6842S + 1.0}$$

$$R_3(s) = \frac{0.158833S^2 - 0.0083}{3.166S^3 + 5.066S^2 + 3.633S + 1}$$

$$R_4(s) = \frac{S^3}{20S^4 + 32S^3 + 24S^2 + 8S + 1.0}$$

The third, and final step in the procedure is to apply what is termed a reciprocal transformation, defined by

$$H_k(s) = \frac{1}{s} x R_k(1/s)$$

which is merely a reversal of the order of the polynomial coefficients. Thus, for the example given, the reduced order approximations are given by the following:

$$H_1(s) = \frac{0.03125}{S + 0.625}$$

$$H_2(s) = \frac{0.056316}{S^2 + 1.6842S + 1.05263}$$

$$H_3(s) = \frac{-0.0083S^2 + 0.158333}{S^3 + 3.633S^2 + 5.066S + 3.166}$$

$$H_4(s) = \frac{1}{S^4 + 8S^3 + 24S^2 + 32S + 20}$$

As expected, the fourth-order approximation is the same as the original equation.

The poles of the approximants are illustrated in table IV.2. The poles of the approximants approach the dominant poles of the original higher-order equation as the order of the approximant is increased.

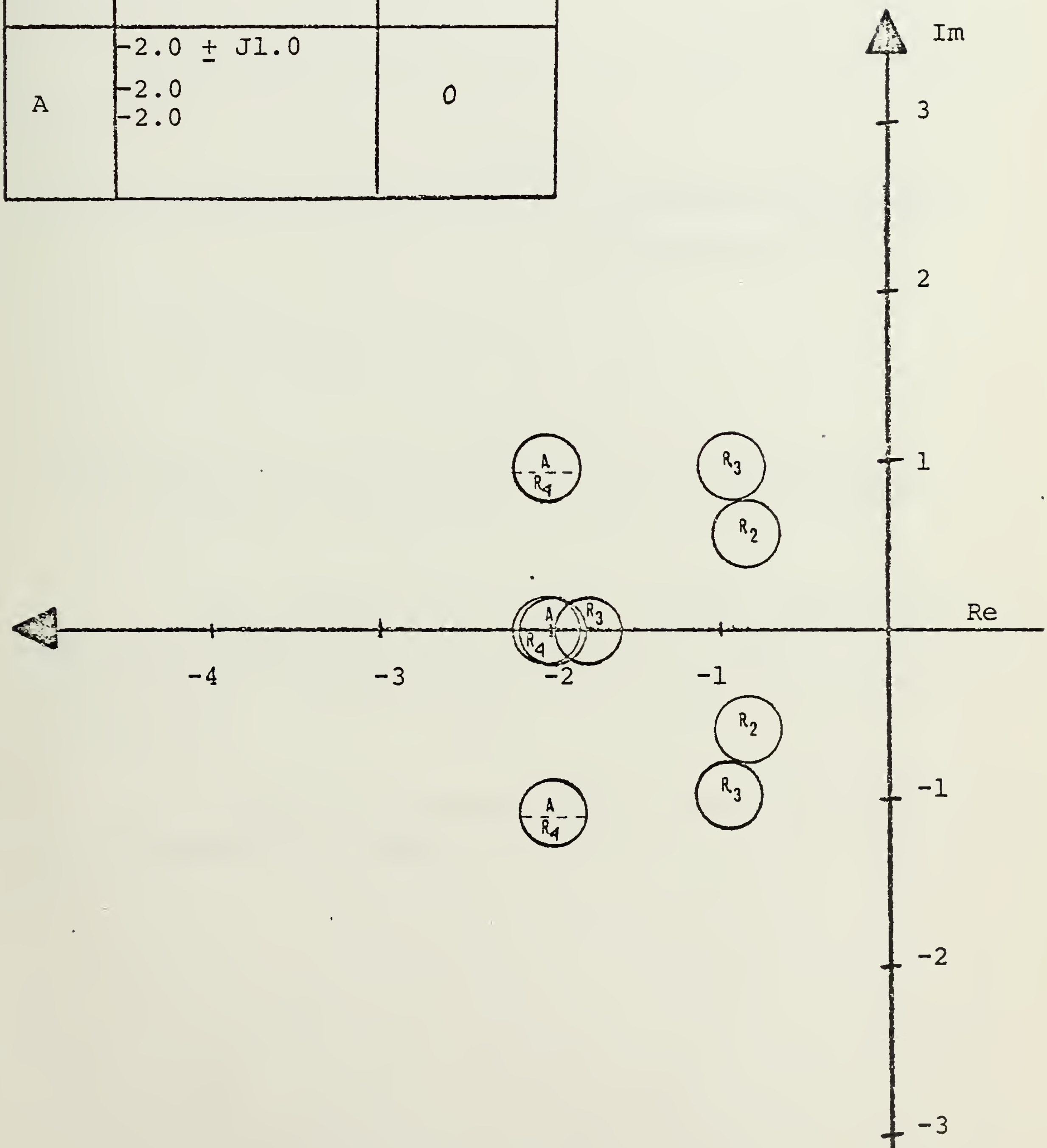
	C_{d1}	C_{d2}	C_{d3}	C_{d4}	...
	C_{d5}	C_{d6}	C_{d7}	...	
$Alpha_1=C_{d1}/C_{d5}$	$W_1=C_{d2}-Alpha_1 C_{d6}$	$W_2=C_{d3}-Alpha_1 C_{d7}$	$W_3=C_{d4}-Alpha_1 C_{d8}$	...	
$Alpha_2=C_{d5}/W_1$	$W_4=C_{d6}-Alpha_2 W_2$	$W_5=C_{d7}-Alpha_2 W_3$	...		
$Alpha_3=W_1/W_4$	$W_6=W_2-Alpha_3 W_5$	$W_7=W_3-Alpha_3$ ...			
$Alpha_4=W_4/W_6$	$W_8=W_5-Alpha_4 W_7$	...			
⋮					

	C_{n1}	C_{n2}	C_{n3}	C_{n4}	...
	C_{n5}	C_{n6}	C_{n7}	C_{n8}	...
$Beta_1=C_{n1}/C_{d5}$	$U_1=C_{n2}-Beta_1 C_{d5}$	$U_2=C_{n3}-Beta_1 C_{d6}$	...		
$Beta_2=C_{n5}/W_1$	$U_3=C_{n6}-Beta_2 W_2$	$U_4=C_{n7}-Beta_2 W_3$	...		
$Beta_3=U_1/W_4$	$U_5=U_2-Beta_3 W_5$	...			
$Beta_4=U_3/W_6$	$U_6=U_4-Beta_4 W_7$	...			
...					

ROUTH TABLE

Table IV.1

Poles and Zeros of Approximant		
Order	Poles	Zeros
R_2	$-0.842 \pm J0.586$	0
R_3	$-0.920 \pm J0.958$ -1.792	± 4.36
R_4	$-2.0 \pm J1.0$ -2.0 -2.0	0
A	$-2.0 \pm J1.0$ -2.0 -2.0	0



POLES OF ROUTH APPROXIMANTS
Table IV.2

V. COMPUTER EXAMPLE

A. SEVENTH ORDER SYSTEM

A seventh order system with the transfer function given by [Ref 2], as

$$G(s)=\frac{384\times10^{7}}{S^{7}+432S^{6}+62670S^{5}+3615900S^{4}+75114000S^{3}+553920000S^{2}\ldots\ldots+1443200000S+384\times10^{7}}$$

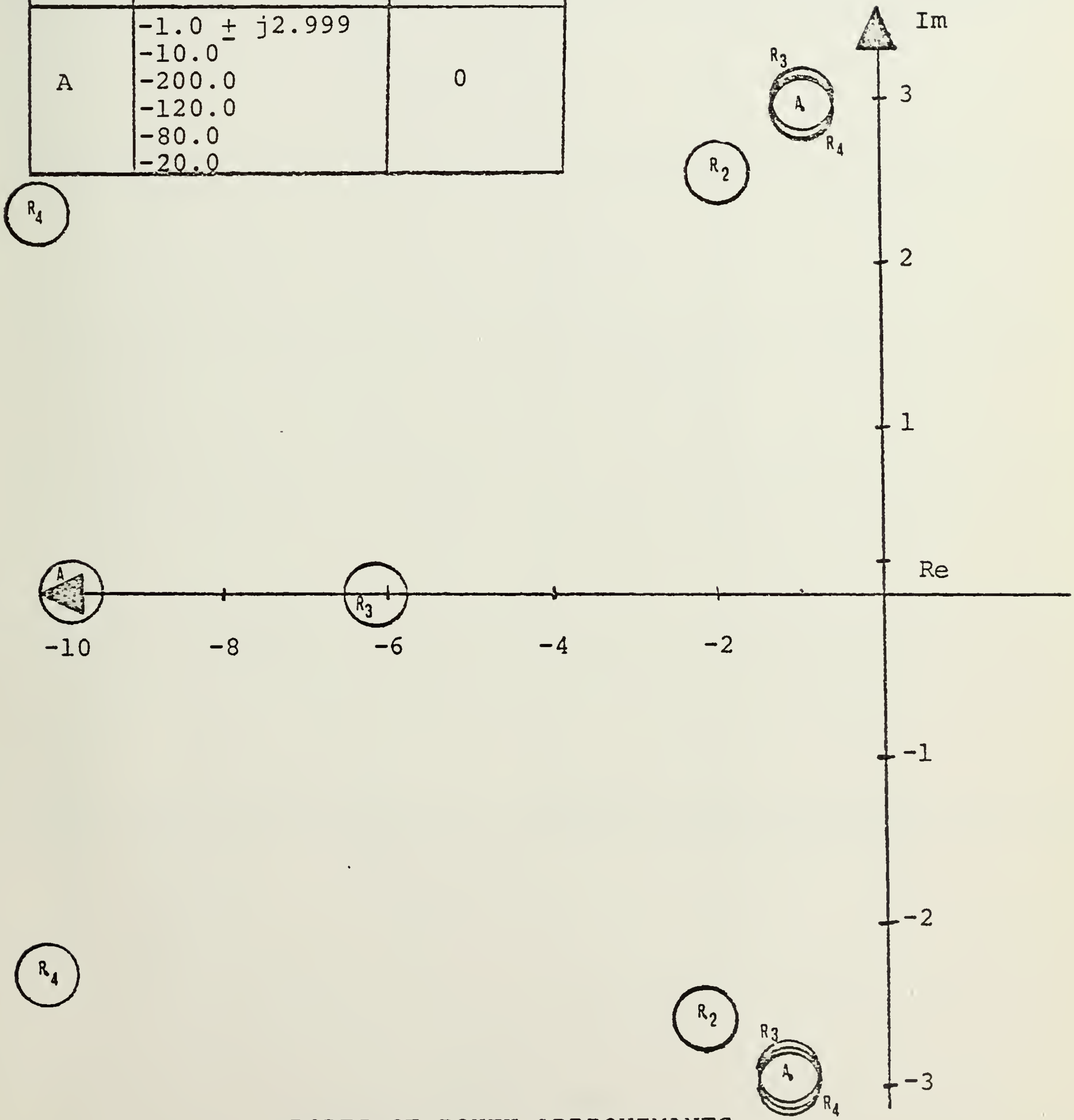
or in factored form as

$$G(s)=\frac{384\times10^{7}}{(S^{2}+2S+10)\ (S+10)\ (S+20)\ (S+80)\ (S+120)\ (S+200)}$$

was reduced to the low order approximants by utilizing ROUTH1 and the response to input excitations, consisting of a Step and Ramp are illustrated in figures 5.1 through 5.6.

The roots of the system and its low order approximants are illustrated in table V.1.

Poles and Zeros of Approximant		
Order	Poles	Zeros
R_2	$-2.038 \pm j2.586$	0
R_3	$-1.084 \pm j2.970$ -6.291	± 10.132
R_4	$-1.0 \pm j2.999$ $-10.89 \pm j2.30$	± 316.58
A	$-1.0 \pm j2.999$ -10.0 -200.0 -120.0 -80.0 -20.0	0



POLES OF ROUTH APPROXIMANTS

Table V.1

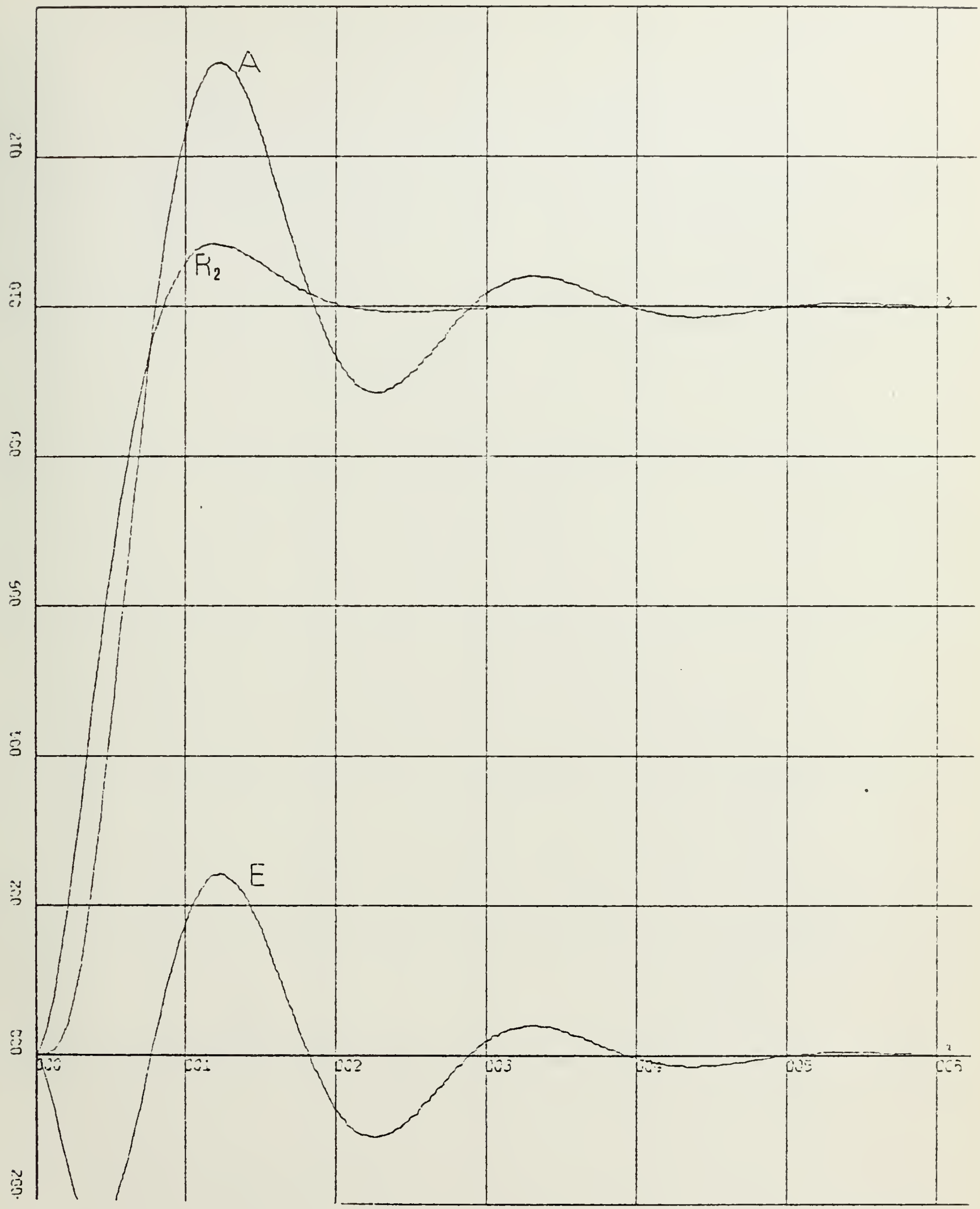


Figure 5.1

STEP INPUT

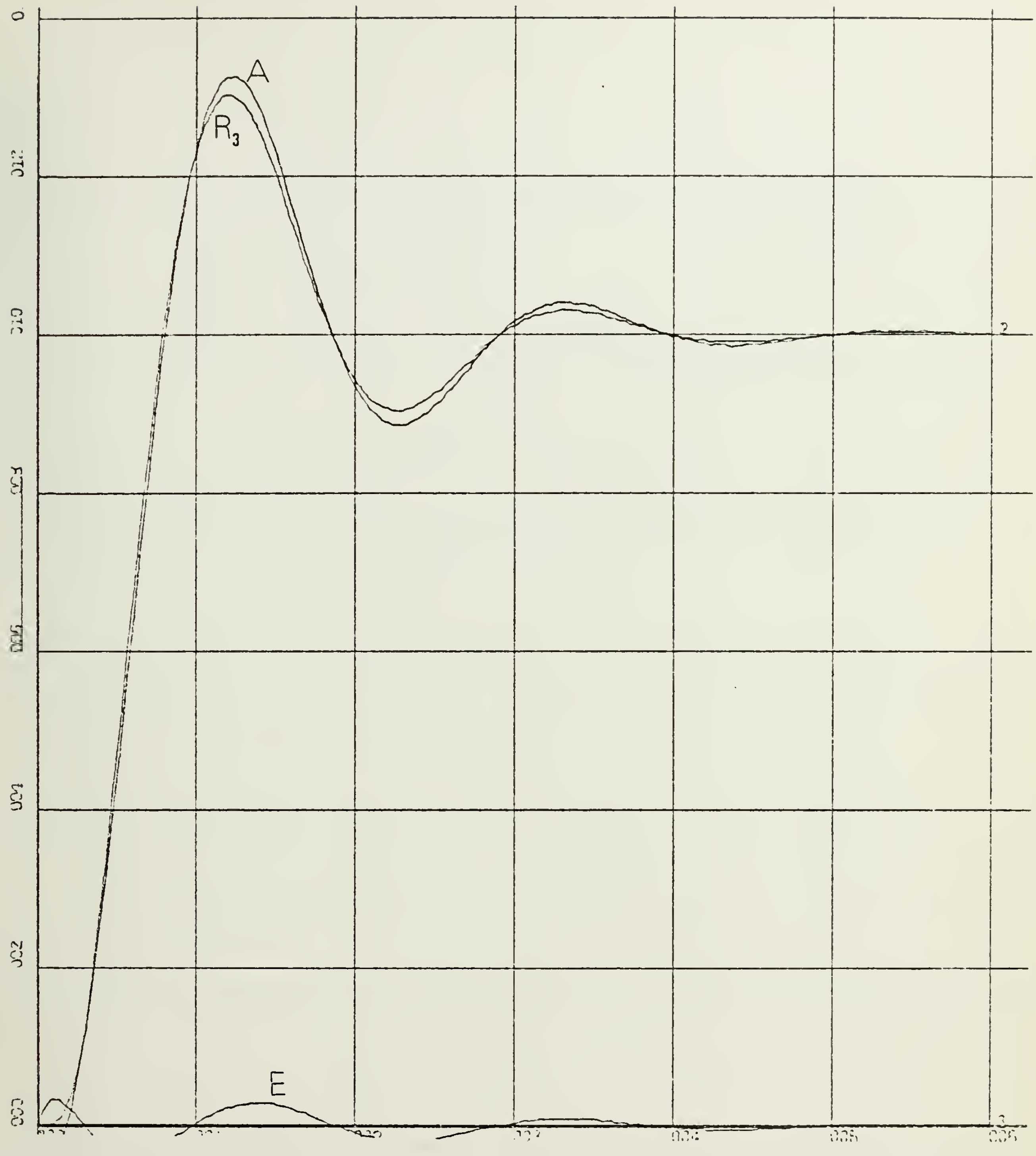


Figure 5.2
STEP INPUT

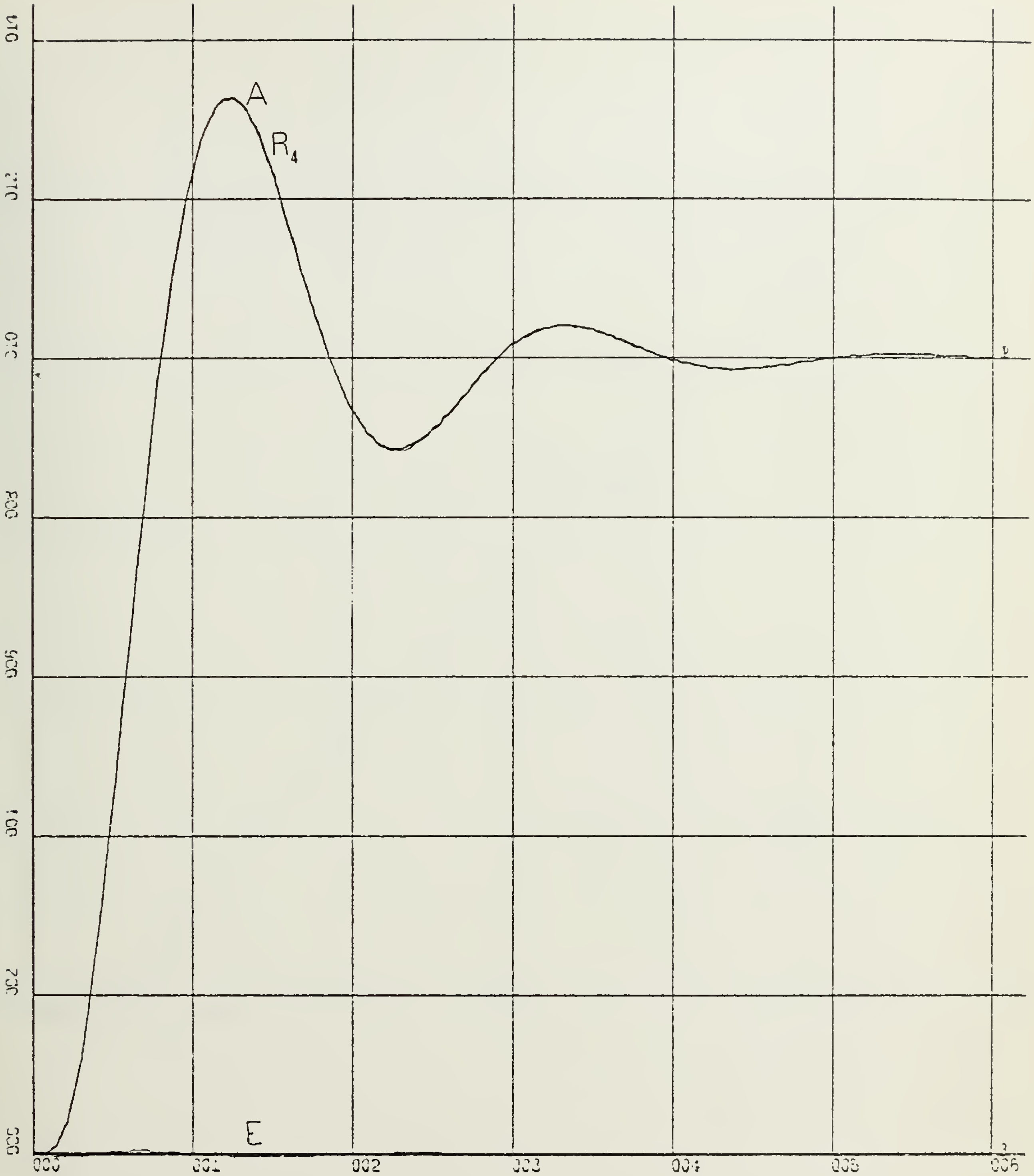


Figure 5.3

STEP INPUT

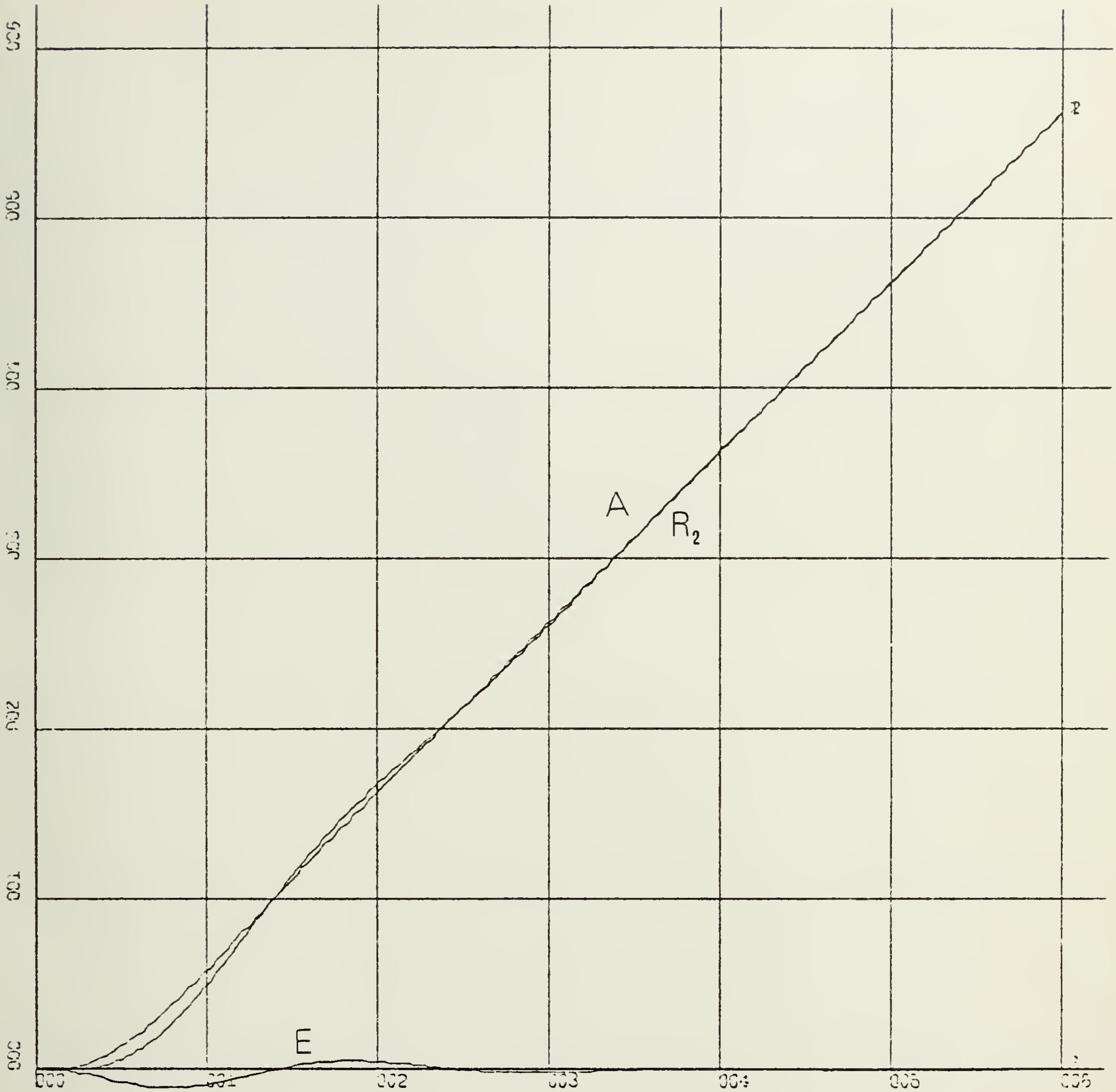


Figure 5.4

RAMP INPUT

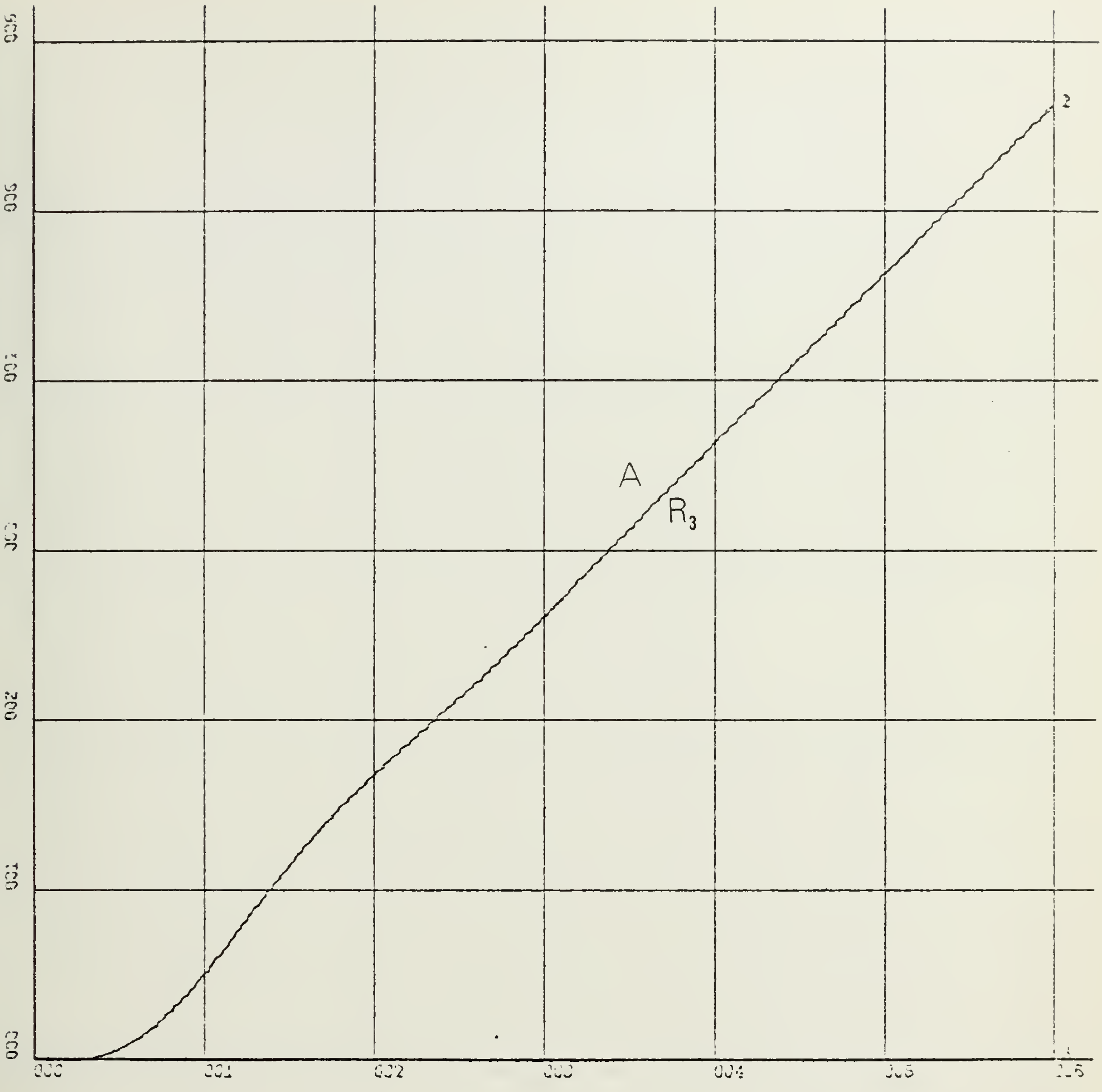


Figure 5.5

RAMP INPUT

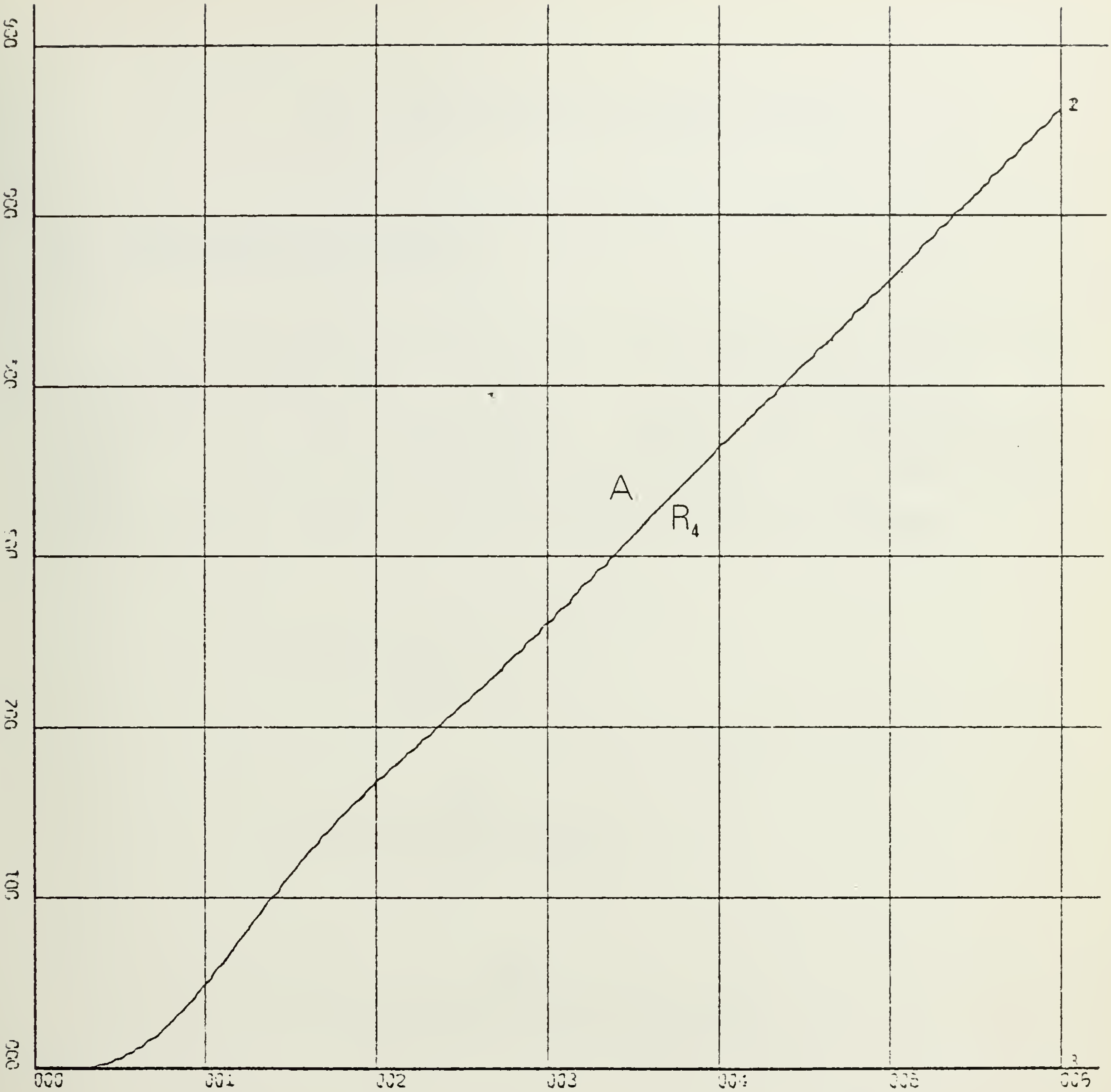


Figure 5.6

RAMP INPUT

VI. COMPARISON TO OTHER METHODS

A. DOMINANT POLE METHOD

The Dominant Pole Approximation method is based upon utilizing the poles closest to the imaginary axis. The equation must be factored to obtain the characteristic roots. For the given seventh-order system, the lower- order equations are given by this method as the following:

$$H(s)=\frac{10}{S^2+2S+10}$$

$$H(s)=\frac{100}{S^3+14S^2+30S+100}$$

$$H(s)=\frac{2000}{S^4+34S^3+310S^2+700S+2000}$$

Graphical plots of the Dominant Pole reduced equations, in response to Step inputs, are illustrated in figures 6.1 through 6.3 in comparison to the original equation and figures 6.4 through 6.6 illustrate the comparison to the Routh equations. The error between the systems is also plotted. Table VI.1 gives the analytical data for performance measure comparisons.

B. ITERATIVE OPTIMIZATION METHOD

The following equations, representing the seventh-order example, were taken from [Ref 2]. This method makes use of an iterative minimization technique to locate the best pole and zero locations for the lower-order models.

$$H(s)=\frac{7.203856}{S^2+1.98616S+7.203856}$$

$$H(s)=\frac{52.2861}{S^3+7.1383S^2+19.5015S+52.2861}$$

$$H(s)=\frac{1470.1403}{S^4+28.5204S^3+209.3842S^2+552.5241S+1470.1403}$$

Plots of these equations versus the Original equation and the Routh equations are illustrated in figures 6.7 through 6.12 in response to Step inputs. Table VI.1 provides numerical data for performance measure comparisons.

C. PRESENTATION OF DATA

Comparisons were made between the "Routh Approximation Method", and that of the Dominant Pole and Iterative Optimization methods previously described.

The basis for comparison consists of the following criteria:

1. Peak Overshoot---$M_{pt}=Y_{max}/Y_{ss}$
2. Delay Time--------T_d=time for Y(t) to reach 0.5 Y_{ss} the very first time.
3. Rise Time---------T_r=time for Y(t) to go from 0.1 to 0.9 of the final value. $T_r=1/BW$.
4. Settling Time----T_s=time at which $Y(t)=Y_{ss}$
5. Graphical Representation in response to Step inputs.
6. Average Error----$J= \left|\Sigma E/t_i\right|$, t_i=integration steps

Table VI.1 illustrates the above comparisons.

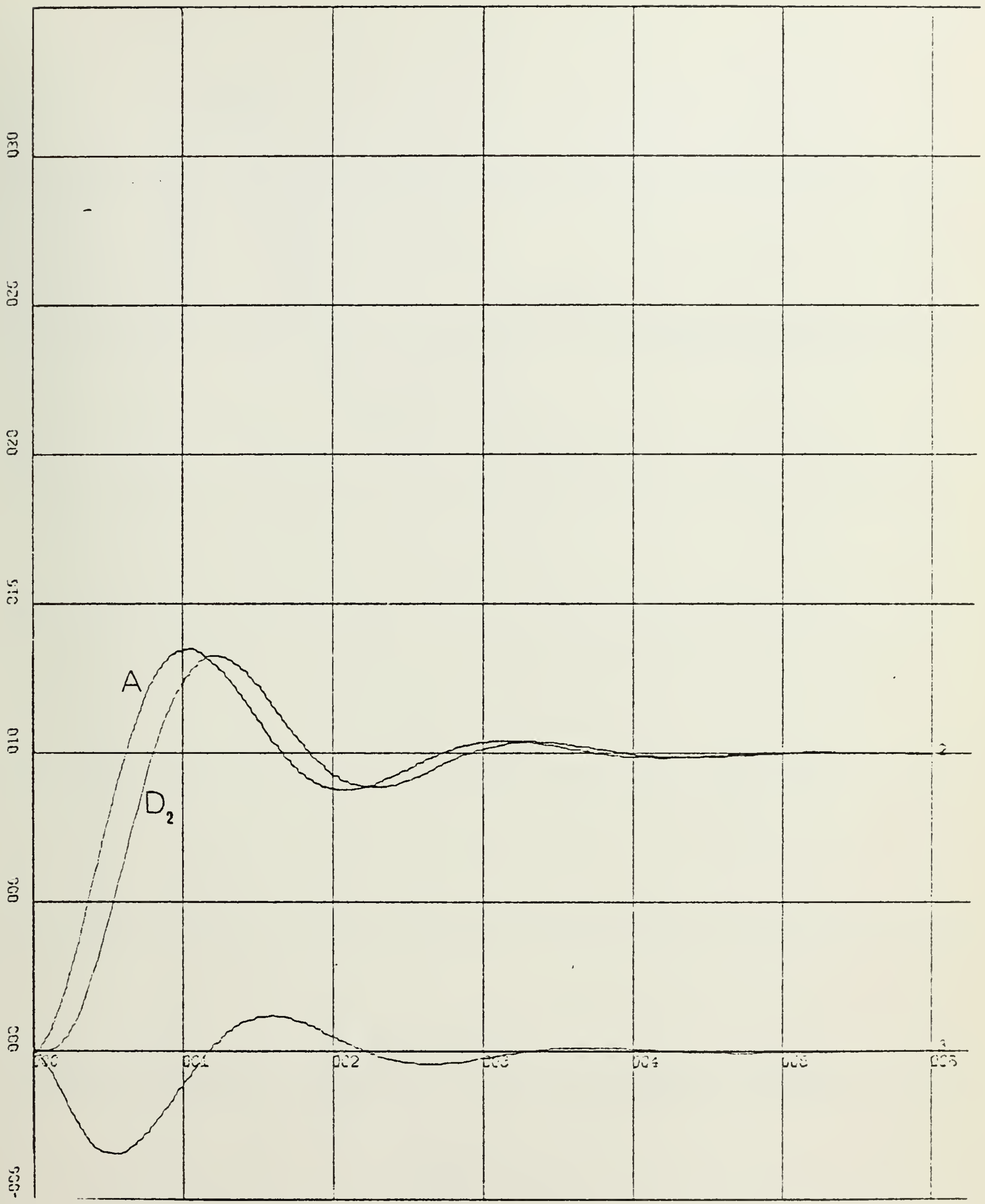


Figure 6.1

STEP INPUT

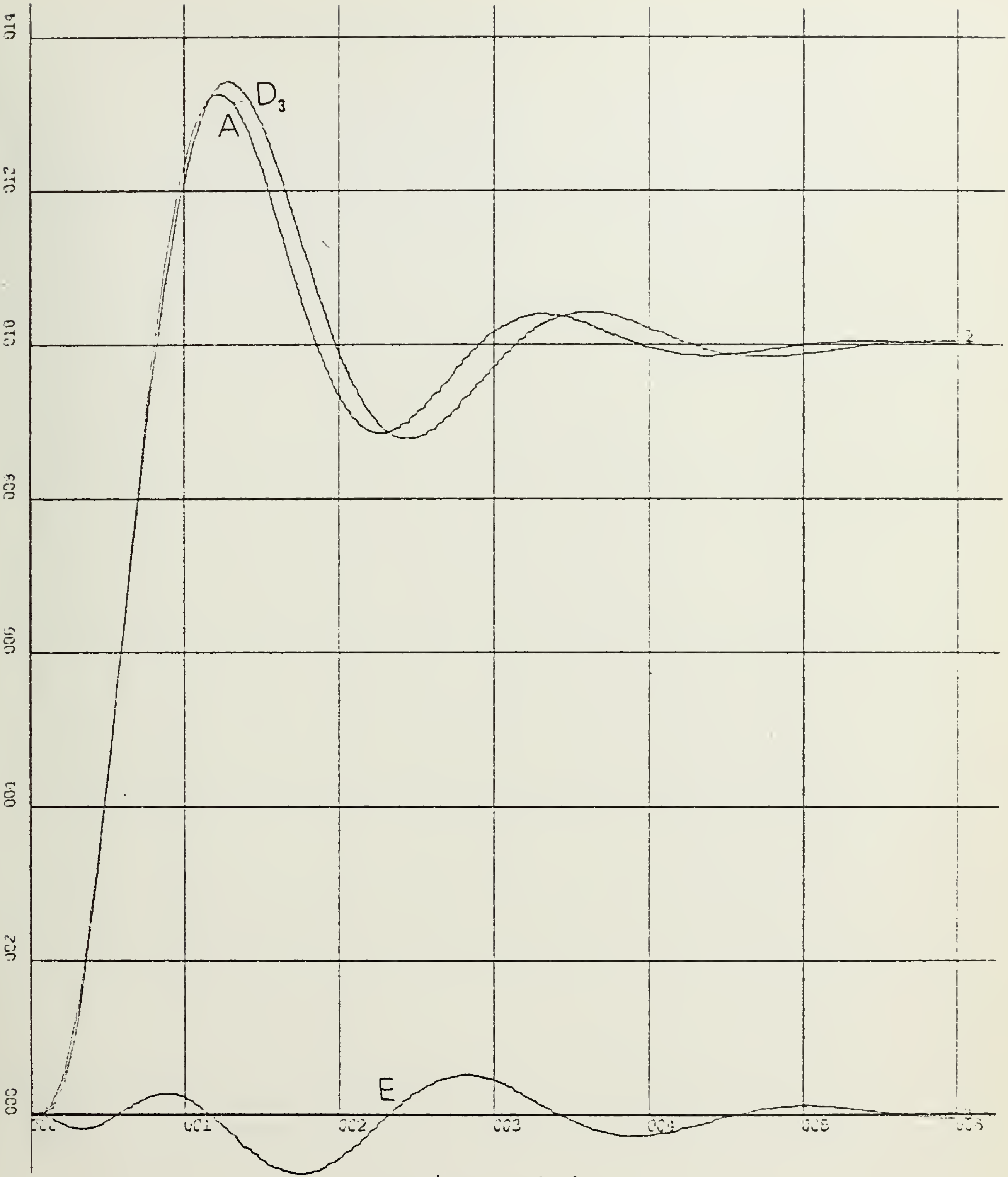


Figure 6 .2

STEP INPUT

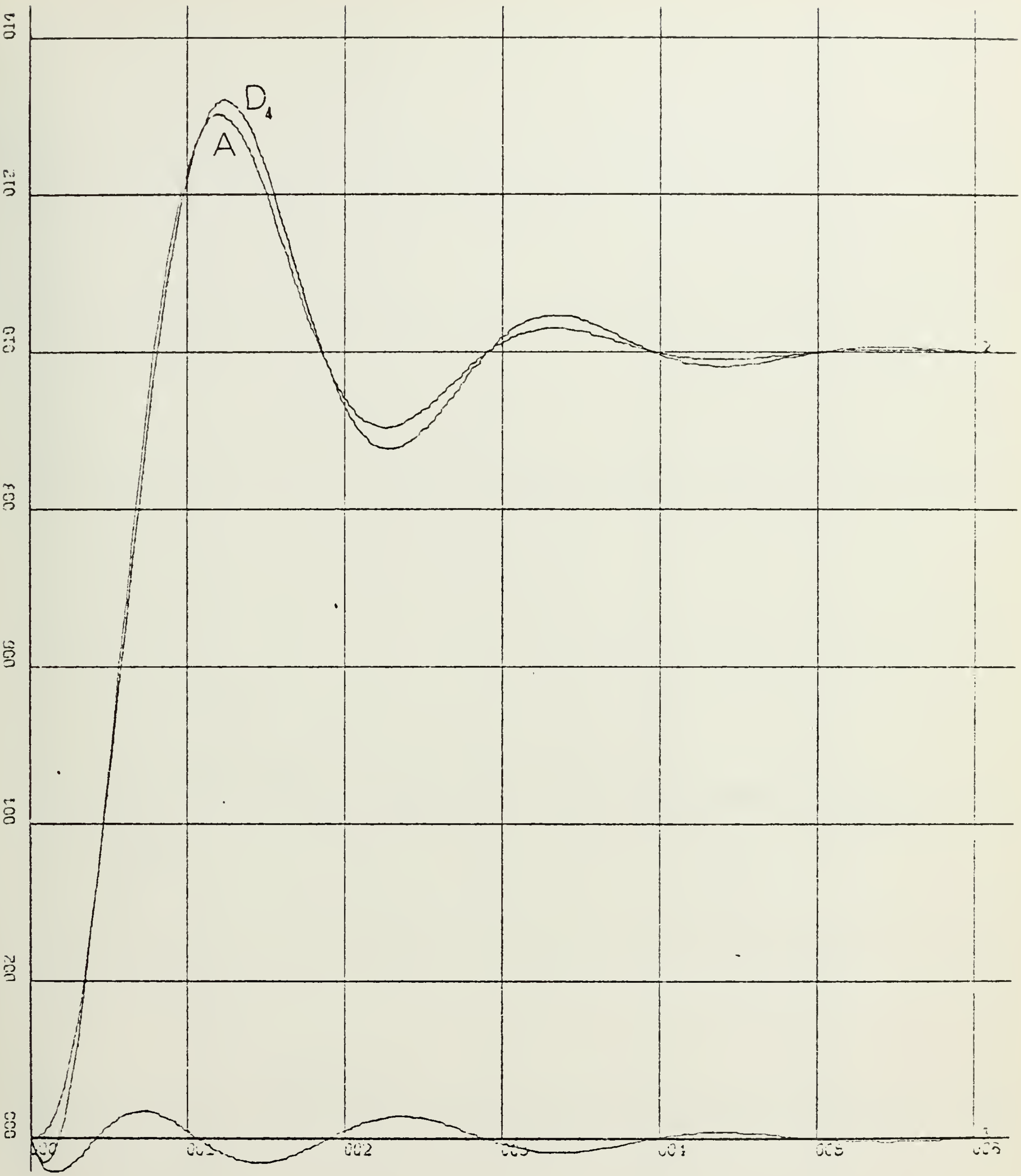


Figure 6 .3

STEP INPUT

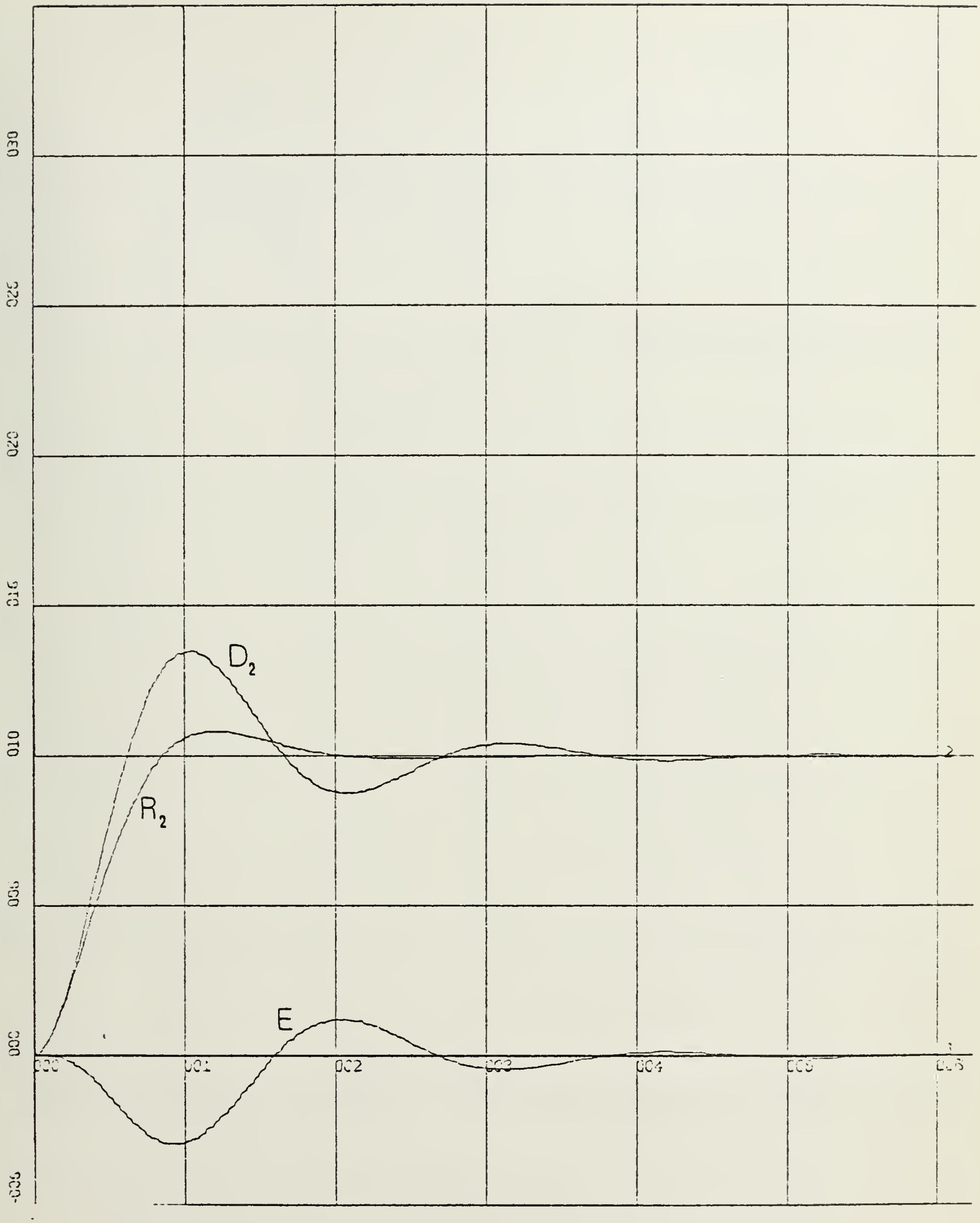


Figure 6 .4

STEP INPUT

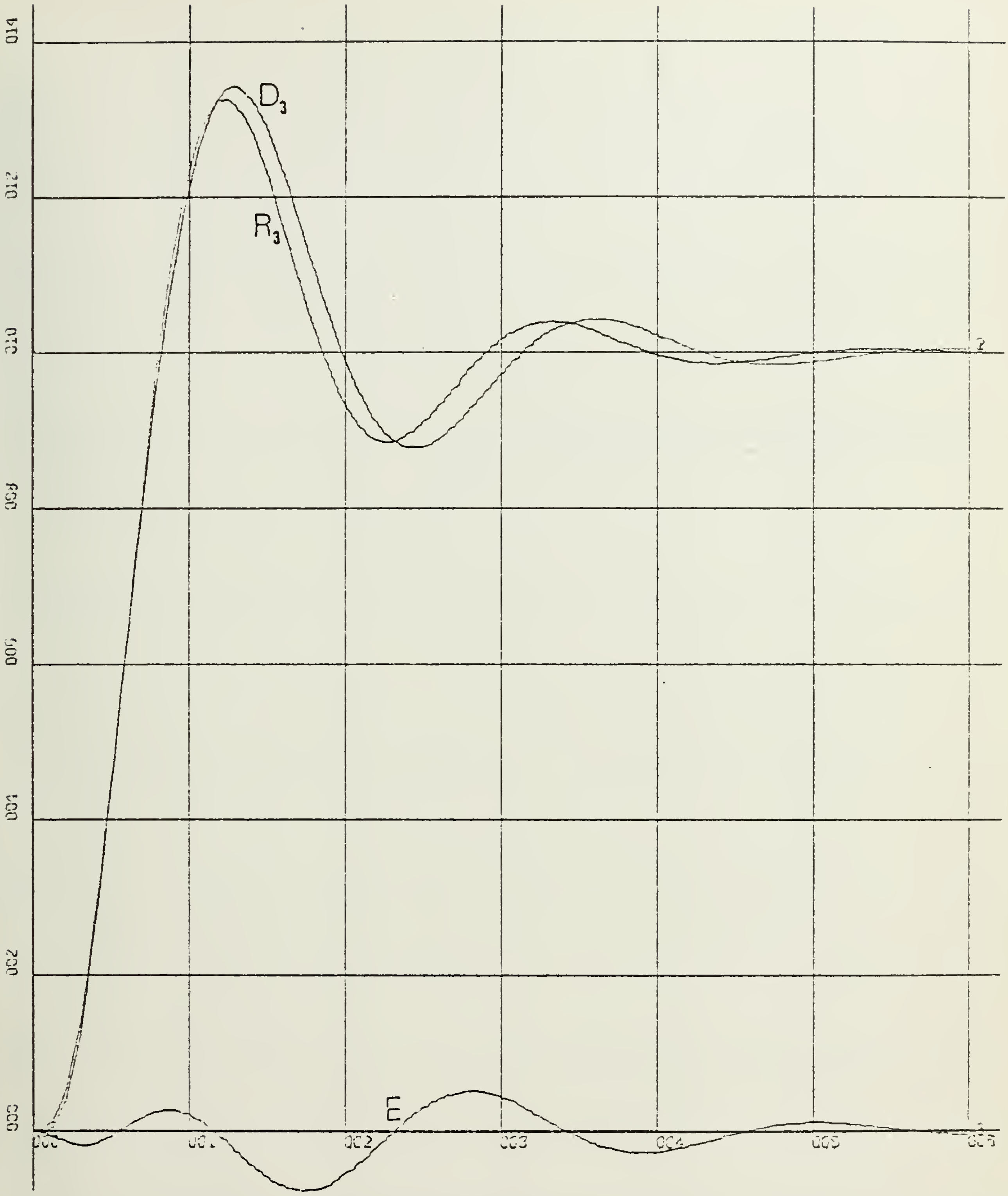


Figure 6 .5

STEP INPUT

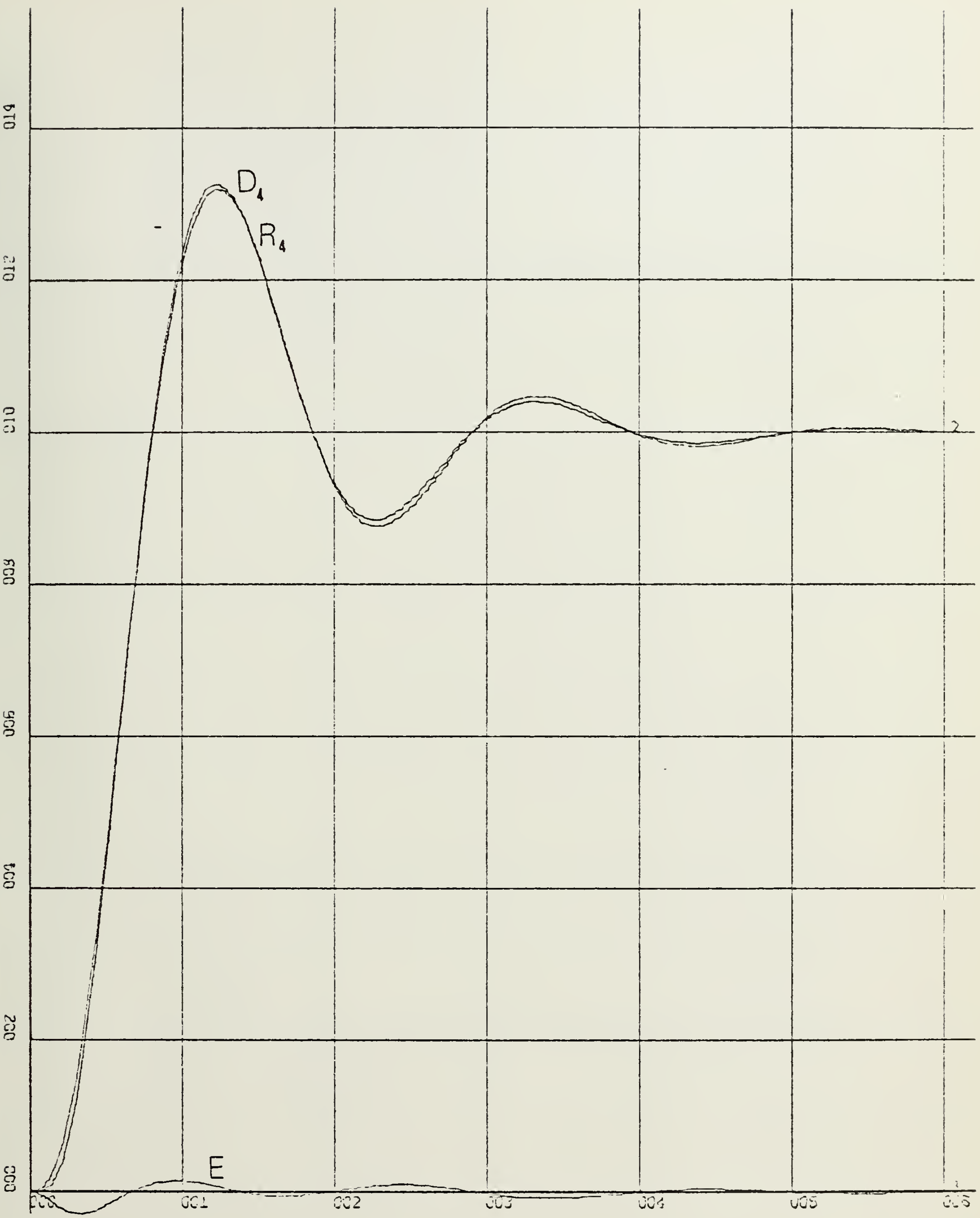


Figure 6 .6
STEP INPUT

Figure 6 .7
STEP INPUT

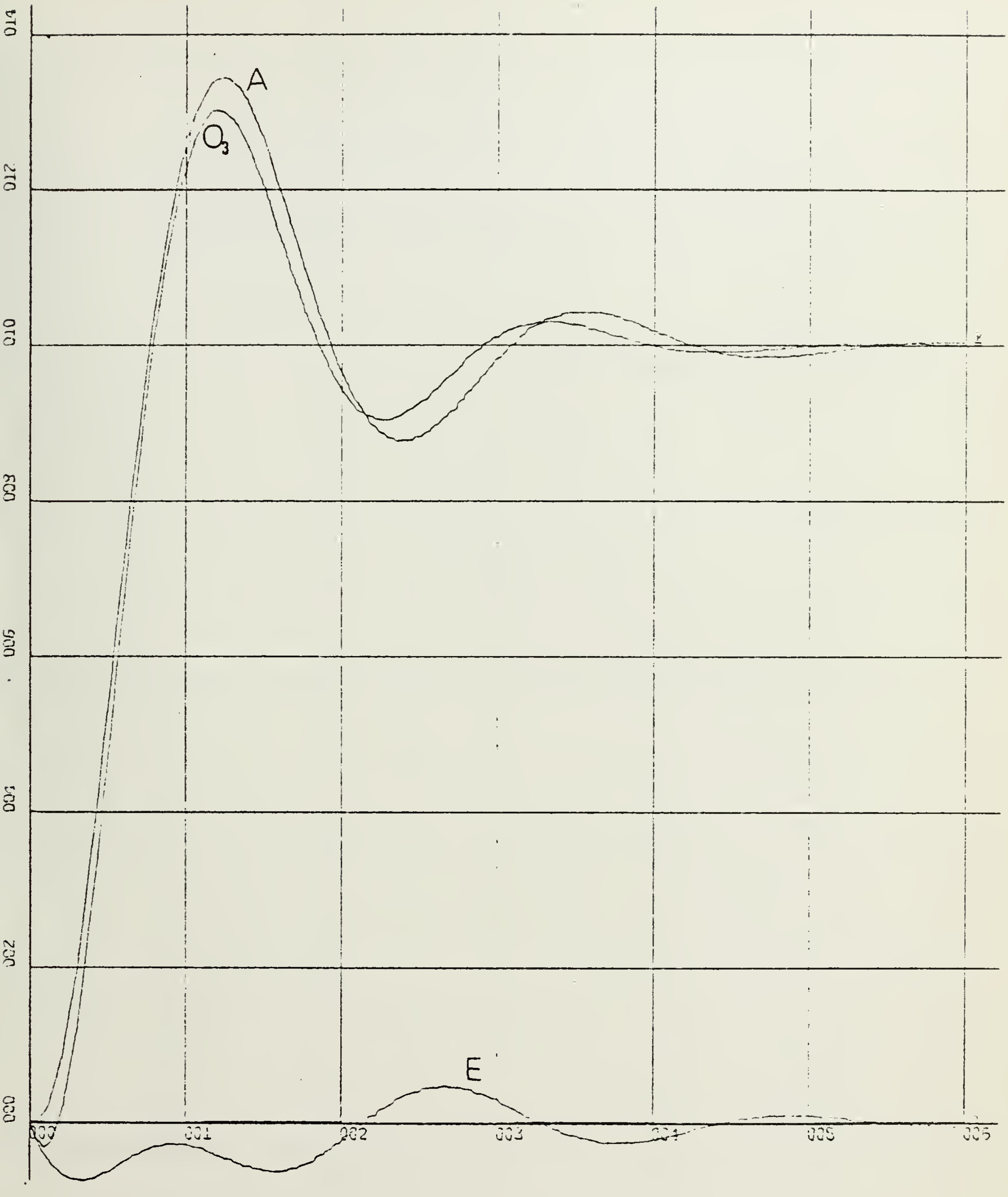


Figure 6.8
STEP INPUT

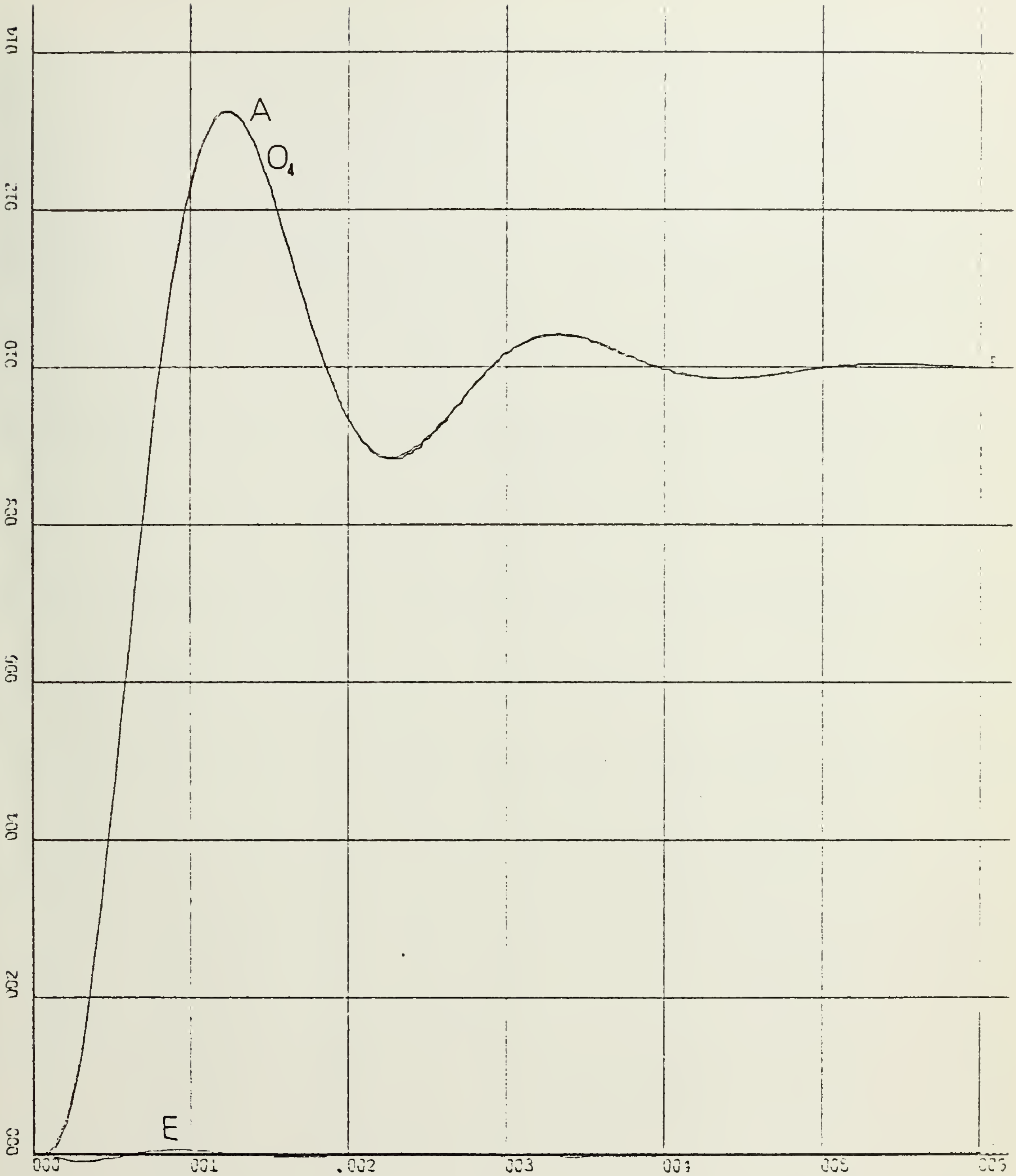


Figure 6.9

STEP INPUT

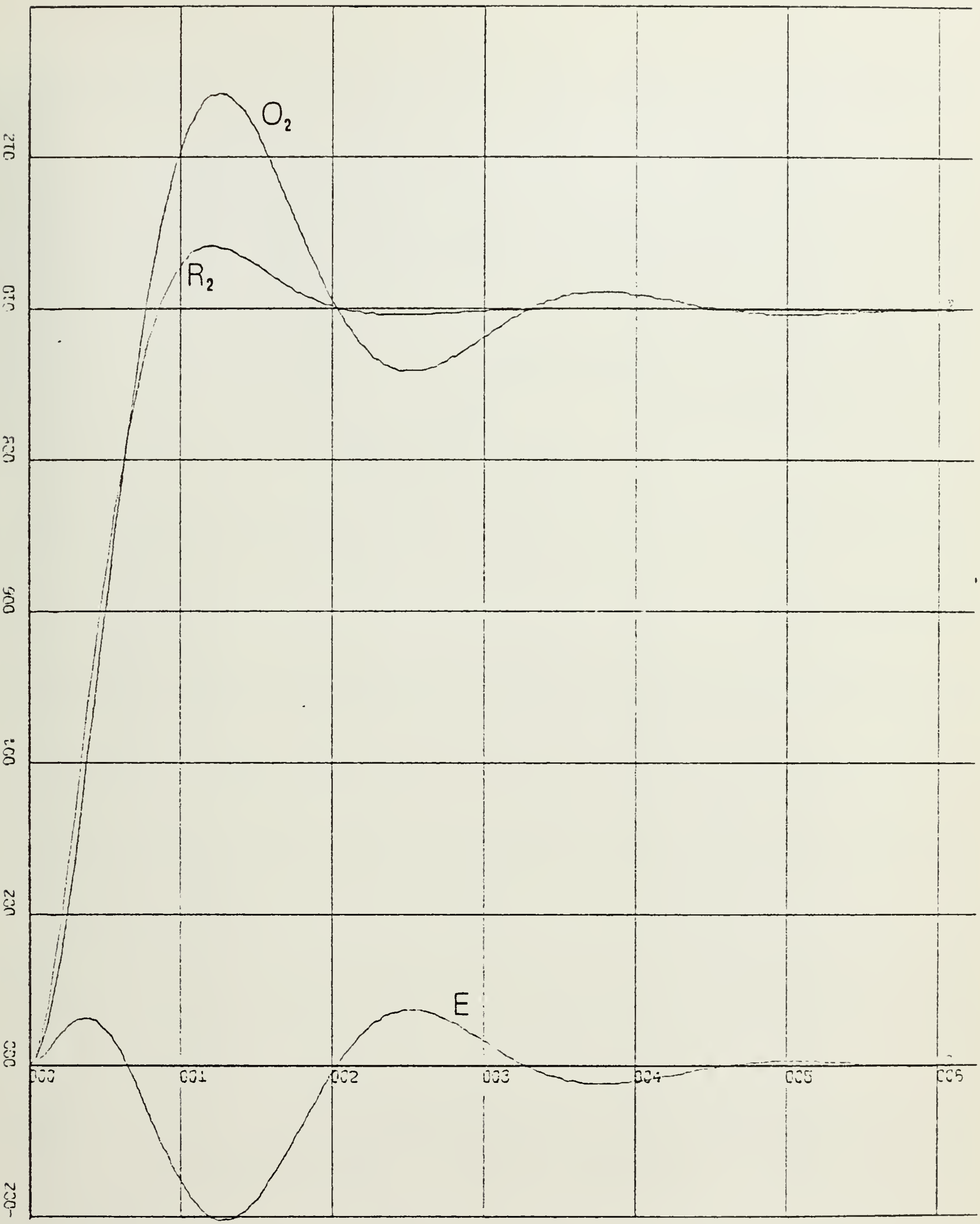


Figure 6 10

STEP INPUT

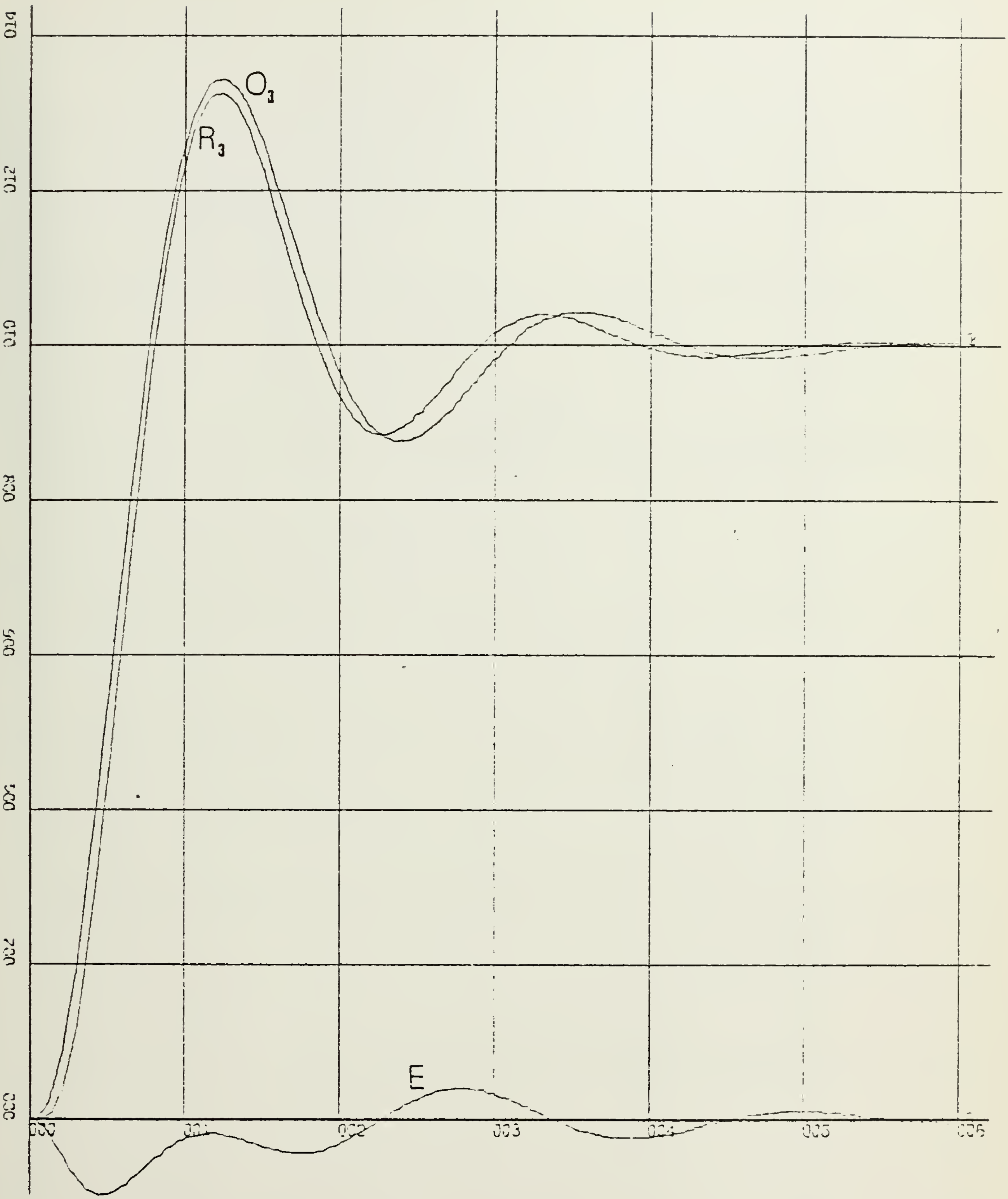


Figure 6 .11

STEP INPUT

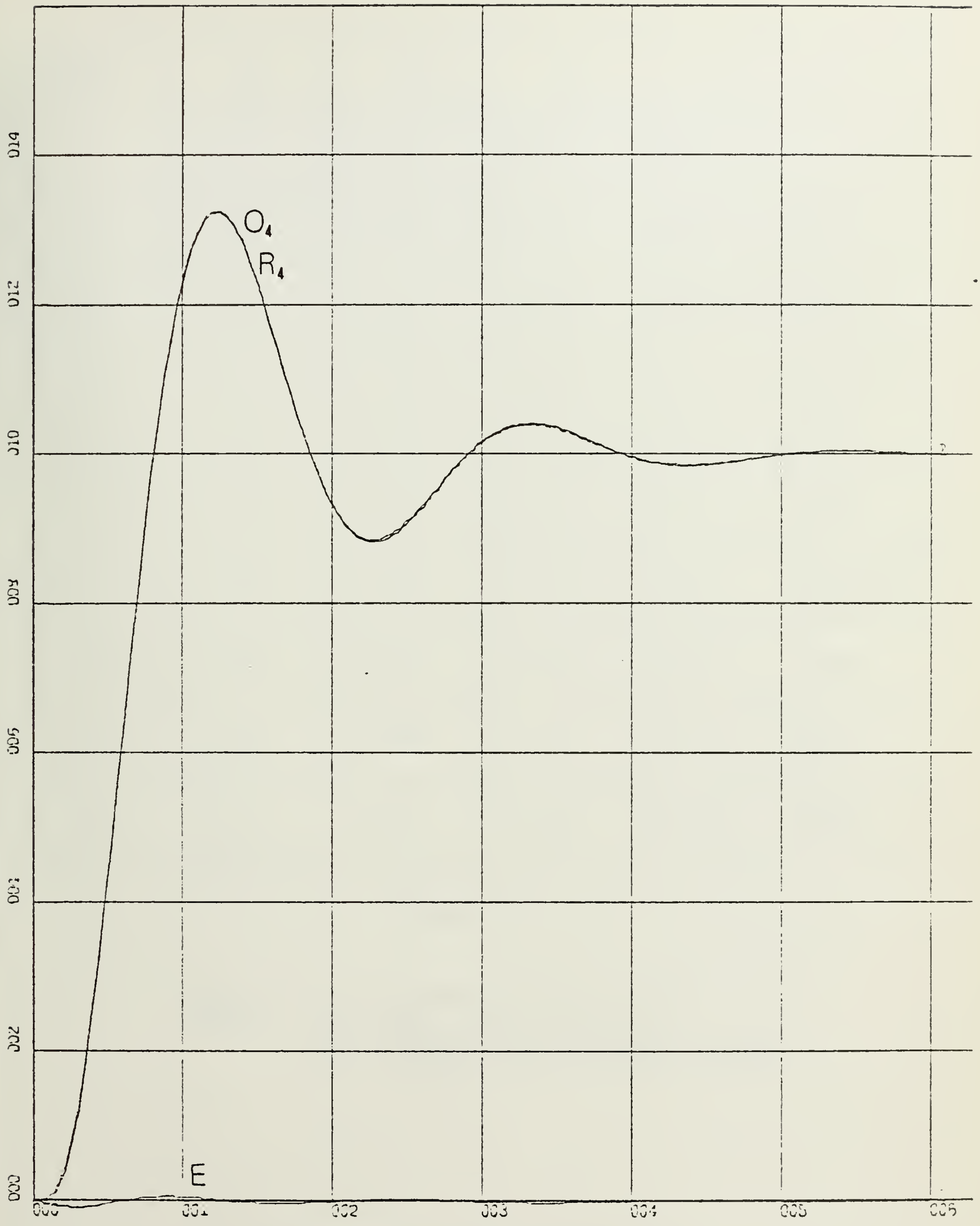


Figure 6.12
STEP INPUT

SYSTEM TYPE	Mpt	Td	Tr	Ts	J
A	1.327	0.533	0.711	5.866	-----
R-4	1.329	0.533	0.711	5.866	$3x10^{-6}$
O-4	1.325	0.533	0.711	5.866	$6x10^{-6}$
D-4	1.338	0.528	0.711	5.866	$1.09x10^{-4}$
R-3	1.247	0.533	0.711	5.866	0.010
O-3	1.320	0.533	0.711	5.866	0.006
D-3	1.346	0.502	0.711	5.866	$1.15x10^{-4}$
R-2	1.083	0.400	0.711	5.866	0.062
O-2	1.285	0.533	0.711	5.866	0.037
D-2	1.350	0.412	0.684	5.866	0.059

TABLE VI.1

Table Symbols
A = Original Seventh Order
R = Routh Approximation
D = Dominant Pole Approximation
O = Optimum Minimization Method

PERFORMANCE MEASURE COMPARISONS

VII. CONCLUSION

The graphical data and tabulated analysis of this thesis indicate that the Routh Approximation Method is a valuable formulation technique which produces very satisfactory results in acquiring approximations to higher-order systems with minimum cost.

In comparison to the other methods of analysis, the Routh Approximation Method offers a quick and easy analytical approach to obtaining low order models. Unlike the Pad'e Approximation, the Routh method ensures stability of the lower-order models, if the original higher-order system is stable. The original system need not be factored, as in the Dominant Pole Approximation method.

The computer program, ROUTH1, utilized to acquire the reduced order equations, the roots of the equations and the graphical plots and numerical tables, takes considerably less time than the minimization technique utilized in reference 2. However, the minimization technique operates efficiently without prior knowledge of the system's transfer function. In the Routh method the availability of the transfer function is a necessary requirement.

In comparing the low order equations of the various methods discussed, the Routh method has proven to be a valid and efficient solution to the problem of obtaining good low order approximants to complex higher-order systems.

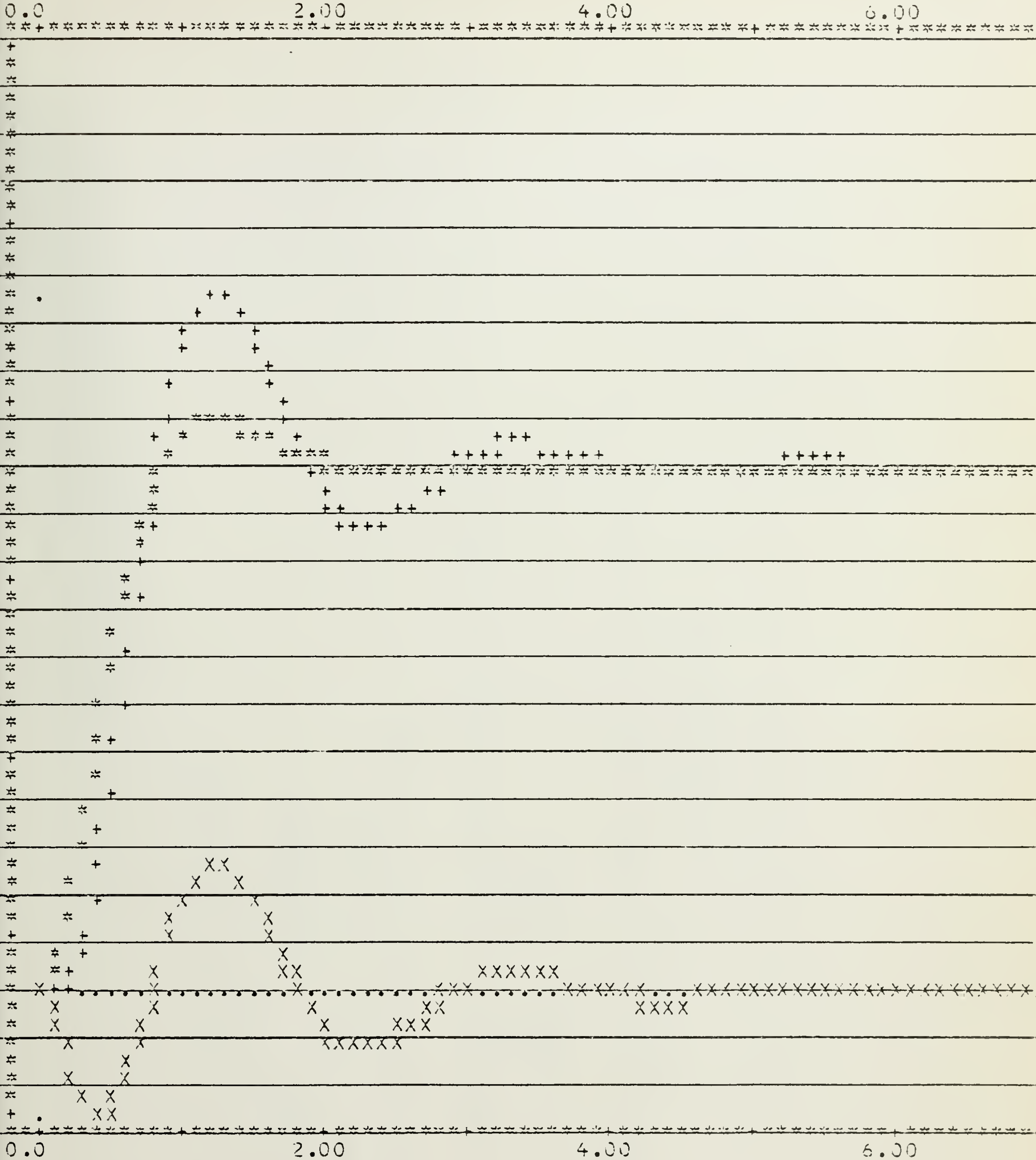
```
REDUCED EQUATIONS IN ASCENDING POWERS OF S

        ORDER ( 1)
NUMER    0.266075E 01
DENOM    0.266075E 01 0.100000E 01
        ORDER ( 2)
NUMER    0.108456E 02 0.0
DENOM    0.108456E 02 0.407514E 01 0.100000E 01
        ORDER ( 3)
NUMER    0.628943E 02 0.0              -0.612693E 00
DENOM    0.628943E 02 0.236378E 02 0.845981E 01 0.100000E 01
        ORDER ( 4)
NUMER    0.124036E 04 0.0              -0.123751E 01 0.0
DENOM    0.124036E 04 0.488169E 03 0.177634E 03 0.237975E 02 0.100000E 01

  ROOTS OF DENOMINATOR OF ORDER 7   ERROR,   0
    REAL PART            IMAG. PART
     -0.200000E 03           0.0
     -0.120000E 03           0.0
     -0.799996E 02           0.0
     -0.199999E 02           0.0
     -0.599991E 01           0.0
     -0.100001E 01           0.299998E 01
     -0.100001E 01          -0.299998E 01

  ROOTS OF DENOMINATOR OF ORDER 2   ERROR,   0
    REAL PART            IMAG. PART
     -0.203807E 01           0.258687E 01
     -0.203807E 01          -0.258687E 01

  ROOTS OF DENOMINATOR OF ORDER 3   ERROR,   0
    REAL PART            IMAG. PART
     -0.629165E 01           0.0
     -0.108407E 01           0.297011E 01
     -0.108407E 01          -0.297011E 01

  ROOTS OF DENOMINATOR OF ORDER 4   ERROR,   0
    REAL PART            IMAG. PART
     -0.108982E 02           0.230245E 01
     -0.108982E 02          -0.230245E 01
     -0.100049E 01           0.299932E 01
     -0.100049E 01          -0.299932E 01

  ERROR CODES
  IER=0    NO ERRORS
  IER=1    NO CONVERGENCE WITH FEASIBLE TOLERANCE
  IER=2    POLY IS DEGENERATE (CONSTANT OR ZERO)
  IER=3    SUBROUTINE ABANDONED (ZERO DIVISOR)
  IER=4    NO S-FRACTION EXISTS
  IER=-1    POOR ACCURACY IN CALCULATIONS
```

COMPUTER OUTPUT

```
                              THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS

                              3 RUNS ARE CALLED FOR

INPUT DATA RECORD FOR RUN NUMBER 1

ORDER OF EQUATIONS = 19
INITIAL TIME       = 0.0
FINAL TIME         = 0.8000E 01
STEP SIZE          = 0.8889E-02

THE ONLY NON-ZERO CONSTANT IS
          C( 1) = 0.1000E 01

ALL THE INITIAL CONDITIONS ARE ZERO

THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

          TIME     X( 0)
          ORIG     X(20)
        SECOND     X(21)
         ERROR     X(24)

THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARE

      POSIT VS TIME        X(20) VS. X( 0)
                           X(21) VS. X( 0)
                           X(24) VS. X( 0)
```

THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS

TIME	ORIG	SECOND	ERROR
0.0	0.0	0.0	0.0
0.17778E 00	0.19033E-01	0.13286E 00	-0.11376E 00
0.35555E 00	0.17644E 00	0.40275E 00	-0.22631E 00
0.53333E 00	0.47702E 00	0.67507E 00	-0.19805E 00
0.71110E 00	0.81911E 00	0.88402E 00	-0.64906E-01
0.88888E 00	0.11050E 01	0.10127E 01	0.92278E-01
0.10666E 01	0.12780E 01	0.10721E 01	0.20597E 00
0.12444E 01	0.13273E 01	0.10837E 01	0.24351E 00
0.14222E 01	0.12770E 01	0.10595E 01	0.21741E 00
0.15999E 01	0.11700E 01	0.10462E 01	0.12378E 00
0.17777E 01	0.10509E 01	0.10239E 01	0.27038E-01
0.19555E 01	0.95434E 00	0.10074E 01	-0.53109E-01
0.21332E 01	0.89847E 00	0.99771E 00	-0.99242E-01
0.23110E 01	0.88561E 00	0.99353E 00	-0.10791E 00
0.24888E 01	0.90623E 00	0.99303E 00	-0.86798E-01
0.26665E 01	0.94497E 00	0.99444E 00	-0.49473E-01
0.28443E 01	0.98636E 00	0.99644E 00	-0.10074E-01
0.30221E 01	0.10189E 01	0.99825E 00	0.20655E-01
0.31998E 01	0.10368E 01	0.99953E 00	0.37267E-01
0.33776E 01	0.10398E 01	0.10003E 01	0.39542E-01
0.35554E 01	0.10316E 01	0.10006E 01	0.31044E-01
0.37331E 01	0.10177E 01	0.10006E 01	0.17095E-01
0.39109E 01	0.10033E 01	0.10004E 01	0.23772E-02
0.40887E 01	0.99239E 00	0.10003E 01	-0.78706E-02
0.42664E 01	0.98671E 00	0.10001E 01	-0.13405E-01
0.44442E 01	0.98616E 00	0.10000E 01	-0.13848E-01
0.46219E 01	0.98936E 00	0.99996E 00	-0.10593E-01
0.47997E 01	0.99436E 00	0.99994E 00	-0.55801E-02
0.49775E 01	0.99931E 00	0.99994E 00	-0.62704E-03
0.51552E 01	0.10030E 01	0.99995E 00	0.30069E-02
0.53330E 01	0.10047E 01	0.99997E 00	0.47802E-02
0.55108E 01	0.10048E 01	0.99998E 00	0.47335E-02
0.56885E 01	0.10035E 01	0.99999E 00	0.35464E-02
0.58663E 01	0.10018E 01	0.99999E 00	0.17658E-02
0.60441E 01	0.10000E 01	0.99999E 00	0.58770E-04
0.62218E 01	0.99883E 00	0.99999E 00	-0.11539E-02
0.63996E 01	0.99823E 00	0.99999E 00	-0.17039E-02
0.65774E 01	0.99834E 00	0.99999E 00	-0.16510E-02
0.67551E 01	0.99880E 00	0.99999E 00	-0.11871E-02
0.69329E 01	0.99943E 00	0.99999E 00	-0.55367E-03
0.71107E 01	0.10000E 01	0.99999E 00	0.31114E-04
0.72884E 01	0.10004E 01	0.99999E 00	0.43261E-03
0.74662E 01	0.10006E 01	0.99999E 00	0.59950E-03
0.76440E 01	0.10005E 01	0.99999E 00	0.56040E-03
0.78217E 01	0.10004E 01	0.99999E 00	0.38633E-03
0.79995E 01	0.10002E 01	0.99999E 00	0.16463E-03



X-SCALE: "*"= 0.100E 00 UNITS

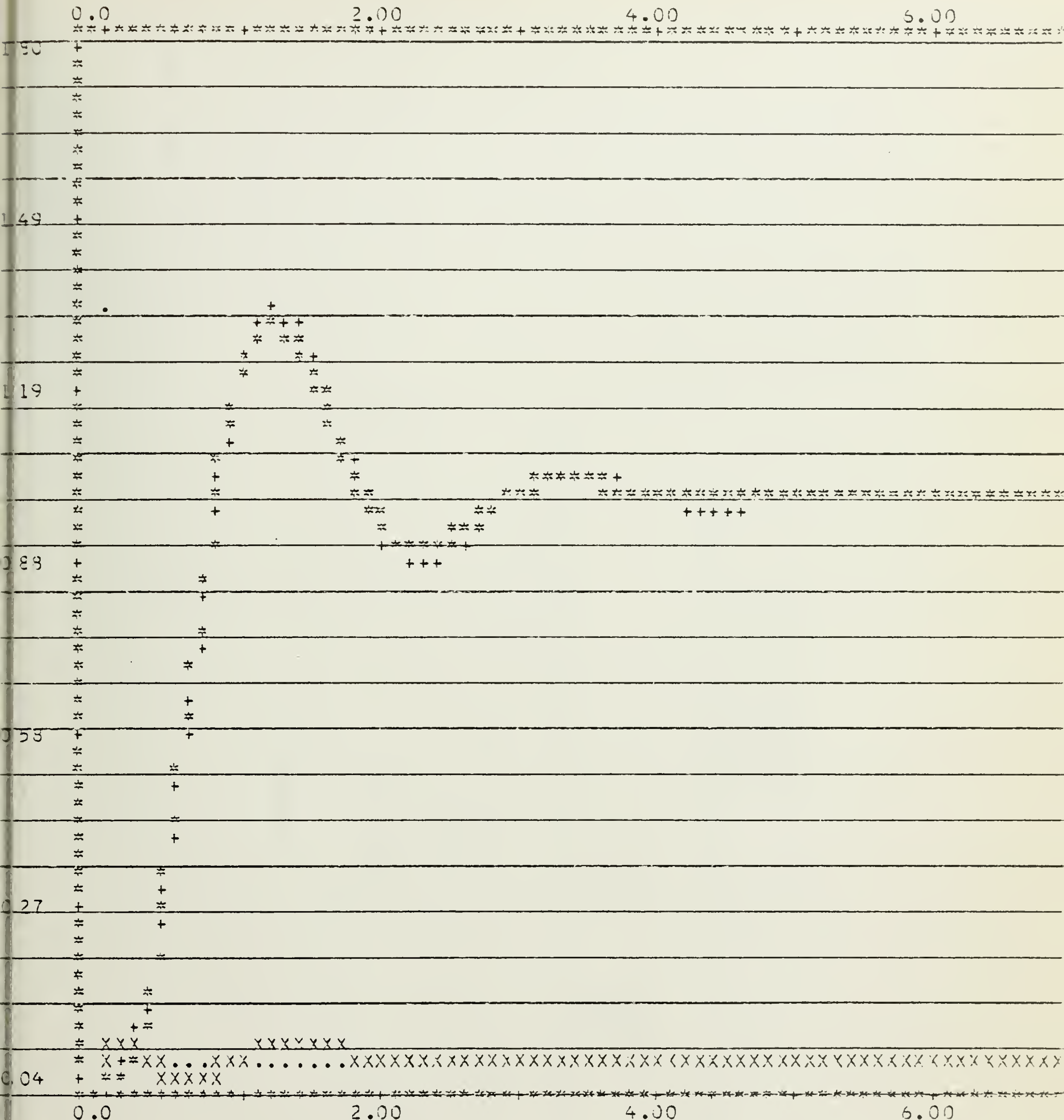
Y-SCALE: "*"= 0.339E-01 UNITS

~~S SEVENTH ORDER VS REDUCED ORDER EQUATIONS RUN 1~~ ~~POSIT VS TIME~~

```
THESIS SEVENTH-ORDER VS REDUCED-ORDER EQUATIONS

3 RUNS ARE CALLED FOR

INPUT DATA RECORD FOR RUN NUMBER 2

ORDER OF EQUATIONS = 19
INITIAL TIME       = 0.0
FINAL TIME         = 0.8000E 01
STEP SIZE          = 0.6889E-02

THE ONLY NON-ZERO CONSTANT IS
           C( 1) = 0.1000E 01

ALL THE INITIAL CONDITIONS ARE ZERO

THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

          TIME     X( 0)
          ORIG     X(20)
         THIRD     X(22)
         ERROR     X(25)

THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARE

      POSIT VS TIME        X(20) VS. X( 0)
                           X(22) VS. X( 0)
                           X(25) VS. X( 0)
```

THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS

TIME	ORIG	THIRD	ERROR
0.0	0.0	0.0	0.0
0.17778E 00	0.19093E-01	-0.83620E-02	0.27455E-01
0.35555E 00	0.17544E 00	0.18853E 00	-0.12085E-01
0.53333E 00	0.47702E 00	0.51171E 00	-0.34690E-01
0.71110E 00	0.81911E 00	0.85034E 00	-0.31228E-01
0.88888E 00	0.11050E 01	0.11170E 01	-0.11942E-01
0.10666E 01	0.12780E 01	0.12678E 01	0.10244E-01
0.12444E 01	0.13273E 01	0.13018E 01	0.25443E-01
0.14222E 01	0.12770E 01	0.12473E 01	0.29153E-01
0.15999E 01	0.11700E 01	0.11478E 01	0.22189E-01
0.17777E 01	0.10509E 01	0.10420E 01	0.89540E-02
0.19555E 01	0.95434E 00	0.95932E 00	-0.49812E-02
0.21333E 01	0.89847E 00	0.91348E 00	-0.15013E-01
0.23110E 01	0.88561E 00	0.90439E 00	-0.18778E-01
0.24888E 01	0.90623E 00	0.92262E 00	-0.16389E-01
0.26665E 01	0.94457E 00	0.95473E 00	-0.93155E-02
0.28443E 01	0.98636E 00	0.98820E 00	-0.18374E-02
0.30221E 01	0.10139E 01	0.10139E 01	0.49620E-02
0.31998E 01	0.10368E 01	0.10279E 01	0.89149E-02
0.33776E 01	0.10398E 01	0.10303E 01	0.95568E-02
0.35554E 01	0.10316E 01	0.10241E 01	0.74644E-02
0.37331E 01	0.10177E 01	0.10138E 01	0.38471E-02
0.39109E 01	0.10033E 01	0.10033E 01	0.46730E-04
0.40887E 01	0.99239E 00	0.99525E 00	-0.28617E-02
0.42664E 01	0.98671E 00	0.99102E 00	-0.43077E-02
0.44442E 01	0.98616E 00	0.99043E 00	-0.42637E-02
0.46219E 01	0.98936E 00	0.99247E 00	-0.31076E-02
0.47997E 01	0.99430E 00	0.99573E 00	-0.14265E-02
0.49775E 01	0.99931E 00	0.99911E 00	0.20200E-03
0.51552E 01	0.10030E 01	0.10016E 01	0.13579E-02
0.53330E 01	0.10047E 01	0.10029E 01	0.18625E-02
0.55108E 01	0.10048E 01	0.10030E 01	0.17433E-02
0.56885E 01	0.10035E 01	0.10023E 01	0.11950E-02
0.58663E 01	0.10018E 01	0.10013E 01	0.47493E-03
0.60441E 01	0.10000E 01	0.10002E 01	-0.13311E-03
0.62218E 01	0.99883E 00	0.99946E 00	-0.62257E-03
0.63996E 01	0.99828E 00	0.99907E 00	-0.78392E-03
0.65774E 01	0.99834E 00	0.99904E 00	-0.70357E-03
0.67551E 01	0.99880E 00	0.99927E 00	-0.46521E-03
0.69329E 01	0.99943E 00	0.99961E 00	-0.17101E-03
0.71107E 01	0.10000E 01	0.99994E 00	0.83208E-04
0.72884E 01	0.10004E 01	0.10002E 01	0.24605E-03
0.74662E 01	0.10006E 01	0.10003E 01	0.29564E-03
0.76440E 01	0.10005E 01	0.10003E 01	0.25272E-03
0.78217E 01	0.10004E 01	0.10002E 01	0.15354E-03
0.79995E 01	0.10002E 01	0.10001E 01	0.40054E-04



X-SCALE: "*"= 0.100E 00 UNITS

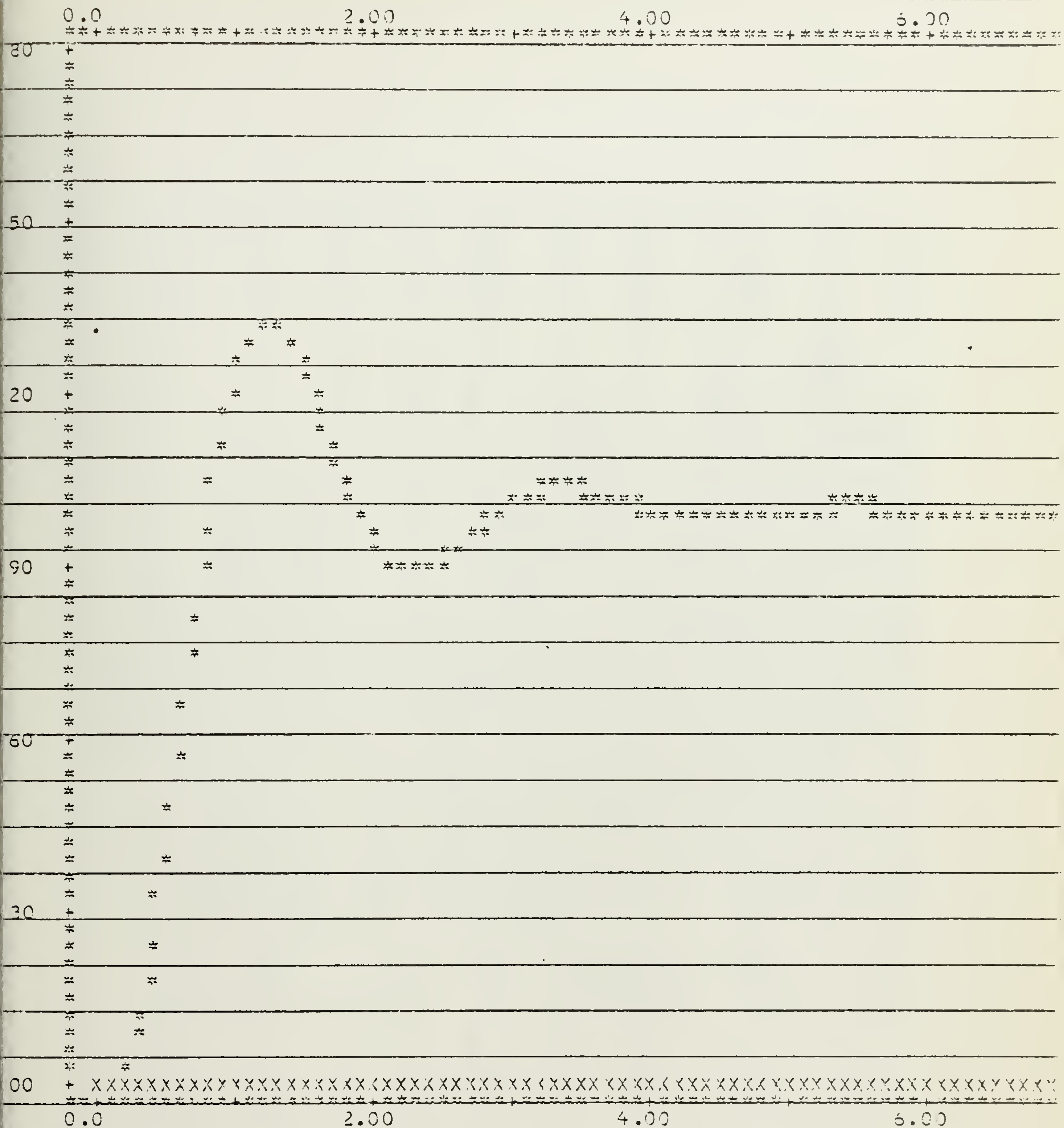
Y-SCALE: "*"= 0.306E-01 UNITS

THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS RUN 2 POSIT VS TIME

```
THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS
3 RUNS ARE CALLED FOR

INPUT DATA RECORD FOR RUN NUMBER 3

ORDER OF EQUATIONS = 19
INITIAL TIME       = 0.0
FINAL TIME         = 0.8000E 01
STEP SIZE          = 0.8889E-02

THE ONLY NON-ZERO CONSTANT IS
         C( 1) = 0.1000E 01

ALL THE INITIAL CONDITIONS ARE ZERO

THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

          TIME    X( 0)
          ORIG    X(20)
        FOURTH    X(23)
         ERROR    X(26)

THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARE

        POSIT VS TIME         X(20) VS. X( 0)
                              X(23) VS. X( 0)
                              X(26) VS. X( 0)
```

THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS

TIME	ORIG	FOURTH	ERROR
0.0	0.0	0.0	0.0
0.17778E 00	0.19093E-01	0.18384E-01	0.70835E-03
0.35555E 00	0.17644E 00	0.17753E 00	-0.10886E-02
0.53333E 00	0.47702E 00	0.47784E 00	-0.82082E-03
0.71110E 00	0.81911E 00	0.81931E 00	-0.20128E-03
0.88888E 00	0.11050E 01	0.11043E 01	0.25272E-03
0.10666E 01	0.12730E 01	0.12776E 01	0.48256E-03
0.12444E 01	0.13273E 01	0.13268E 01	0.49496E-03
0.14222E 01	0.12770E 01	0.12765E 01	0.34714E-03
0.15999E 01	0.11700E 01	0.11698E 01	0.11826E-03
0.17777E 01	0.10509E 01	0.10511E 01	-0.10681E-03
0.19555E 01	0.95434E 00	0.95460E 00	-0.26017E-03
0.21332E 01	0.89647E 00	0.89677E 00	-0.30959E-03
0.23110E 01	0.88561E 00	0.88587E 00	-0.25809E-03
0.24888E 01	0.90623E 00	0.90638E 00	-0.14710E-03
0.26665E 01	0.94497E 00	0.94499E 00	-0.28253E-04
0.28443E 01	0.98636E 00	0.98629E 00	0.71704E-04
0.30221E 01	0.10189E 01	0.10188E 01	0.12779E-03
0.31998E 01	0.10368E 01	0.10367E 01	0.13161E-03
0.33776E 01	0.10398E 01	0.10397E 01	0.95367E-04
0.35554E 01	0.10316E 01	0.10316E 01	0.39101E-04
0.37331E 01	0.10177E 01	0.10177E 01	-0.17166E-04
0.39109E 01	0.10033E 01	0.10034E 01	-0.59128E-04
0.40887E 01	0.99239E 00	0.99247E 00	-0.78201E-04
0.42664E 01	0.98671E 00	0.98673E 00	-0.73314E-04
0.44442E 01	0.98616E 00	0.98622E 00	-0.54240E-04
0.46219E 01	0.98936E 00	0.98939E 00	-0.27537E-04
0.47997E 01	0.99436E 00	0.99436E 00	-0.36359E-05
0.49775E 01	0.99931E 00	0.99930E 00	0.11981E-04
0.51552E 01	0.10030E 01	0.10029E 01	0.19073E-04
0.53330E 01	0.10047E 01	0.10047E 01	0.17166E-04
0.55108E 01	0.10048E 01	0.10048E 01	0.76294E-05
0.56885E 01	0.10035E 01	0.10035E 01	-0.57220E-05
0.58663E 01	0.10018E 01	0.10018E 01	-0.13120E-04
0.60441E 01	0.10000E 01	0.10001E 01	-0.24796E-04
0.62218E 01	0.99883E 00	0.99886E 00	-0.27359E-04
0.63996E 01	0.99828E 00	0.99831E 00	-0.22292E-04
0.65774E 01	0.99834E 00	0.99835E 00	-0.17285E-04
0.67551E 01	0.99880E 00	0.99881E 00	-0.11683E-04
0.69329E 01	0.99943E 00	0.99944E 00	-0.59009E-05
0.71107E 01	0.10000E 01	0.10000E 01	-0.38147E-05
0.72884E 01	0.10004E 01	0.10004E 01	-0.38147E-05
0.74662E 01	0.10006E 01	0.10006E 01	-0.47684E-05
0.76440E 01	0.10005E 01	0.10006E 01	-0.76294E-05
0.78217E 01	0.10004E 01	0.10004E 01	-0.95367E-05
0.79995E 01	0.10002E 01	0.10002E 01	-0.10490E-04



X-SCALE: "*"= 0.100E 00 UNITS

Y-SCALE: "*"= 0.300E-01 UNITS

ESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS RUN 3 POSIT VS TIME

```
C       ************************************************************************
C
C                                 DATA CARCS
C
C       ************************************************************************
C       CARC COLUMN NR              DESCRIPTION                      FORMAT   *
C        1      1-20  PROBLEM IDENTIFICATION                          5A4,2I2  *
C              21-22  IREQ=1 REDUCEC EQUATIONS ONLY                            *
C                     IREQ=2 REDUCED EQN'S + TIME RESPONSE                     *
C              23-24  L=ORDER OF ORIGINAL EQUATION                             *
C       ************************************************************************
C        2      1-12  NUMERATOR COEFFICIENTS IN ASCENDING             4E12.5   *
C              13-24  POWERS OF S. MUST HAVE NUMBER OF                         *
C                     CARDS TO COVER ORDER OF EQUATION                         *
C               ETC                                                            *
C       ************************************************************************
C        3      1-12  DENOMINATOR COEFFICIENTS IN ASCENDING           4E12.5   *
C              13-24  POWERS OF S.                                             *
C               ETC                                                            *
C       ************************************************************************
C        4      1-12  FREQ=FREQUFNCY DESIGNATED FOR SINUSOIDAL         E12.5   *
C                     INPUT IN TIME RESPONSE.                                  *
C       ************************************************************************
C        5      1-48  GRAPH TITLE.  APPEARS ON GRAPHS.                  10A8   *
C       ************************************************************************
C        6       1    NUMBER OF RUNS TO BE PROCESSEC (.LE.9)            I1     *
C                     RUN NUMBER IS PLACEC ON OUTPUT                           *
C       ************************************************************************
C        7      1-10  INITIAL VALUE OF TIME      (ALL PUNCHED         8F10.4   *
C              11-20  INTEGRATION STEP SIZE      WITH DECIMAL                  *
C              21-30  FINAL VALUE OF TIME        POINTS)                       *
C       ************************************************************************
C        8      1-10  VALUES OF CONSTANTS                              8F10.4  *
C              11-20  C(1)=1.0 WILL GIVE STEP INPUT OF 1.0                     *
C               ETC   C(2)=1.0 WILL GIVE RAMP INPUT OF T                       *
C                     C(3)=1.0 WILL GIVE SINUSOIDAL INPUT CF                   *
C                          SIN(OMEGA * T) WHERE OMEGA=2*PI*FREQ                *
C       ************************************************************************
C        9      1-10  INITIAL CONDITION VALUES OF VARIABLES            8F10.4  *
C              11-20  PUNCHED WITH DECIMAL POINTS                              *
C               ETC   X(20)=ORIG EQN, X(21)=SECOND CRDER                       *
C                     X(22)=THIRD ORDER, AND X(23)=FOURTH ORDER                *
C                     USE THREE BLANK CARDS IF ALL IC'S ARE ZERO.              *
C       ************************************************************************
C       10      1-10  CHOICE OF VARIABLES FOR PRINT OUT IN                     *
C              11-20  TABLE FORM.  EACH GROUP OF 10 COLUMNS MAY                *
C               ETC   SPECIFY A TABLE HEADING (8 CHAR) AND A                   *
C                     2 DIGIT, RIGHT JUSTIFIED SUBSCRIPT TO          8(A8,I2)  *
```

```
C                  BE PRINTED UNDER 'ORIG' VS 'TIME'.                  *
C                  8 VARIABLES MAXIMUM FOR ANY ONE RUN.                *
C                     FOR THIS PROGRAM, THE FOLLOWING WILL             *
C                  GIVE THE ORIGINAL AND REDUCED ORDER EQN'S           *
C                  VS TIME FOR RUNS 1, 2, AND 3:                       *
C                  1  TIME 00   ORIG 20 SECOND 21  ERROR 24            *
C                  2  TIME 00   ORIG 20 THIRD  22  ERROR 25            *
C                  3  TIME 00   ORIG 20 FOURTH 23  ERROR 26            *
C     ******************************************************************
C      11     1-20 CHOICE OF VARIABLES FOR GRAPHICAL OUTPUT 4(2A8,2I2)*
C            21-40 UP TO 4 CURVES MAY BE PLOTTED,SEPARATELY            *
C            41-60 OR ALL ON ONE GRAPH.                                *
C             ETC  EX: POSIT VS TIME 2000 IN COL(1-20) WILL            *
C                  CAUSE X(20) TO BE PLOTTED AGAINST TIME              *
C                  AND LABELED ORIG VS TIME.  THE FIRST 16             *
C                  COLUMNS FOR LABEL AND COL'S(17-18) FOR              *
C                  Y COORDINATE AND 19-20 FOR X COORDINATE.            *
C                     FOR MULTIPLE PLOTS ON SINGLE GRAPH,              *
C                  ONLY THE FIRST TITLE REQ'D.                         *
C                     FOR MULTIPLE RUNS ONLY THE DATA CARDS            *
C                  NUMBERS 7 THROUGH 11 NEED TO BE SUPPLIED            *
C                  FOR EACH ADDITIONAL RUN..............................
C
C     ******************************************************************
C       NOTE: INTEGRATION STEP SIZE IS A GUESSTIMATE
C        AND BEST FOUND BY TRIAL AND ERROR.                            *
C           BLANK CARDS MUST BE INCLUDED IN DATA                       *
C        DECK FOR INITIAL CONDITIONS  IF THEY ARE TO BE ZERO.          *
C        INITIAL CONDITION STATE VALUES ARE:                           *
C            ORIG EQN: X(1)--X(10) CORRESPOND TO X(1)-X(10)            *
C        SECOND ORDER: X(11)-X(12) CORRESPOND TO X(1)-X(2)             *
C         THIRD ORDER: X(13)-X(15) CORRESPOND TO X(1)-X(3)             *
C        FOURTH ORDER: X(16)-X(19) CORRESPOND TO X(1)-X(4)             *
C                                                                      *
C                 PROGRAMER: J. D. THOMPSON                            *
C                            LT,USN JUNE 76                            *
C     ******************************************************************
C
      DIMENSION A(4),B(4),W(12),Y(12),U(8),Z(9),X(26),D(15),CP(12),
     1XDOT(19),C(15),CN(12),CD(12),RTR(20),RTI(20),NAME(5),POL(12)
      C(10)=1.
  777 FORMAT('1')
  901 FORMAT(/6X,30HROOTS OF DENOMINATOR OF ORDER ,I1,2X,7HERROR, I2,/8X
     *,9HREAL PART,8X,10HIMAG. PART)
  907 FORMAT(9X,E13.6,7X,E13.6)
 2001 FORMAT(5A4,2I2)
 2002 FORMAT(4E12.5)
 2003 FORMAT (1P6E20.7)
```

```
 2009 FORMAT (6X,23HPROBLEM IDENTIFICATION ,5X,5A4)
 5007 FORMAT(///,4X,42HREDUCED EQUATIONS IN ASCENDING POWERS OF S ,/)
 5008 FORMAT(E12.5/)
 5010 FORMAT(/10X,7HORDER (,I2,1H))
 5011 FORMAT(/4X,7HNUMER   ,6E13.6)
 5012 FORMAT(E12.5)
 5013 FORMAT(/4X,7HDENOM   ,6E13.6)
      READ(5,2001,END=10)(NAME(I),I=1,5),IREQ,L
      WRITE(6,777)
      PRINT 2009,(NAME(I),I=1,5)
      DO 996 I=1,4
      A(I)=0.
  996 B(I)=0.
      DO 997 I=1,12
      W(I)=0.
  997 Y(I)=0.
      DO 998 I=1,11
      CN(I)=0.
  998 CD(I)=0.
C     READ IN NUMERATOR AND DENOMINATOR COEFFICIENTS
C     IN ASCENDING POWERS OF S.
  207 NP=L+1
      READ(5,2002)(CN(I),I=1,NP)
      READ(5,2002)(CD(I),I=1,NP)
      READ(5,5012)FREQ
      PI=3.1416
      OMEGA=2*PI*FREQ
      PRINT 5014
 5014 FORMAT(6X,40HORIGINAL EQUATION, ASCENDING POWERS OF S)
      WRITE(6,5010)L
      WRITE(6,5011)(CN(I),I=1,NP)
      WRITE(6,5013)(CD(I),I=1,NP)
C
C     COMPUTATION OF ALPHA TERMS FOLLOWS
C     BETA COMPUTATIONS
C
      A(1)=CD(1)/CD(2)
      B(1)=CN(1)/CD(2)
      DO 1 I=1,5
      J=I+I+1
      K=J+1
      W(I)=CD(J)-A(1)*CD(K)
    1 Y(I)=CN(J)-B(1)*CD(K)
      W(5)=CD(11)
      Y(5)=CN(11)
      A(2)=CD(2)/W(1)
      B(2)=CN(2)/W(1)
      DO 2 I=6,9
```

```
      K=I-4
      J=K+K
      W(I)=CD(J)-A(2)*W(K)
    2 Y(I)=CN(J)-B(2)*W(K)
      A(3)=W(1)/W(6)
      B(3)=Y(1)/W(6)
      IF(L-4)140,3,3
    3 DO 4 I=10,12
      K=I-3
      J=K-5
      W(I)=W(J)-A(3)*W(K)
    4 Y(I)=Y(J)-B(3)*W(K)
      A(4)=W(6)/W(10)
      B(4)=Y(6)/W(10)
      GO TO 140
  140 Z(1)=1.
      Z(2)=A(1)*A(2)
      Z(3)=A(1)+A(3)
      Z(4)=A(2)*A(3)
      Z(5)=Z(2)*A(3)
      Z(6)=A(2)+A(4)
      Z(7)=Z(2)+A(4)*Z(3)
      Z(8)=A(4)*Z(4)
      Z(9)=A(1)*Z(8)
      U(1)=A(2)*B(1)
      U(2)=B(1)+B(3)
      U(3)=A(3)*B(2)
      U(4)=Z(4)*B(1)
      U(5)=B(2)+B(4)
      U(6)=B(1)*(A(2)+A(4))+B(3)*A(4)
      U(7)=A(3)*A(4)*B(2)
      U(8)=Z(8)*B(1)
      PRINT 5007
      ICRD=1
      PRINT 5010,IORD
      WRITE(6,5011)B(1)
      WRITE(6,5013)A(1),Z(1)
      ICRD=2
      PRINT 5010,IORD
      WRITE(6,5011)U(1),B(2)
      WRITE(6,5013)Z(2),A(2),Z(1)
      ICRD=3
      PRINT 5010,IORD
      WRITE(6,5011)U(4),U(3),U(2)
      WRITE(6,5013)Z(5),Z(4),Z(3),Z(1)
      ICRD=4
      PRINT 5010,IORD
      WRITE(6,5011)U(8),U(7),U(6),U(5)
```

```
      WRITE(6,5013)Z(9),Z(8),Z(7),Z(6),Z(1)
      KOUNT=0
      LC=L+1
      DO 899 I=1,12
  899 CP(I)=CC(I)
      GO TO 900
  900 KOUNT=KOUNT + 1
      CALL PRQD(CP,LD,RTR,RTI,POL,IR,IER)
      IF(KOUNT - 4)299,301,302
  299 IF(KOUNT - 2)300,302,303
  300 KORD=L
      WRITE(6,901)KORD,IER
      DO 77 I=1,L
   77 PRINT 907,RTR(I),RTI(I)
      LC=3
      CP(1)=Z(2)
      CP(2)=A(2)
      CP(3)=Z(1)
      GO TO 900
  302 KORD=2
      WRITE(6,901)KORD,IER
      DO 78 I=1,2
   78 PRINT 907,RTR(I),RTI(I)
      LC=4
      CP(1)=Z(5)
      CP(2)=Z(4)
      CP(3)=Z(3)
      CP(4)=Z(1)
      GO TO 900
  303 KORD=3
      WRITE(6,901)KORD,IER
      DO 79 I=1,3
   79 PRINT 907,RTR(I),RTI(I)
      LC=5
      CP(1)=Z(9)
      CP(2)=Z(8)
      CP(3)=Z(7)
      CP(4)=Z(6)
      CP(5)=Z(1)
      GO TO 900
  301 KORD=4
      WRITE(6,901)KORD,IER
      DO 81 I=1,4
   81 PRINT 907,RTR(I),RTI(I)
      IF(IREQ - 2)800,801,801
C
C     THIS CONCLUDES CALCULATIONS OF NUMERATORS AND DENOMINATORS
C      FOR THE APPROXIMANTS, INCLUDING PRINT OUT OF VALUES
```

```
C       IN ACCENDING POWERS OF S AND IN INCREASING ORDER
C       OF APPROXIMANT.
  800 STOP
   10 GO TO 801
  801 CONTINUE
      PRINT 5015
 5015 FORMAT(/6X,11HERROR CODES,/6X,16HIER=0  NO ERRORS,/6X,45HIER=1  NO
     1 CONVERGENCE WITH FEASIBLE TOLERANCE,/6X,44HIER=2  POLY IS DEGENER
     2ATE (CONSTANT OR ZERO),/6X,42HIER=3  SUBROUTINE ABANDONED (ZERO DI
     3VISOR),/6X,27HIER=4  NO S-FRACTION EXISTS,/6X,37HIER=-1  POOR ACCU
     4RACY IN CALCULATIONS)
      DO 499 I=1,15
  499 D(I)=0
      IF(L-4)500,501,502
  502 IF(L-6)503,504,505
  505 IF(L-8)506,507,508
  508 IF(L-10)509,510,511
  500 D(9)=1
      GO TO 444
  501 D(10)=1
      D(1)=1
      GO TO 444
  503 D(11)=1
      D(4)=1
      D(5)=1
      GO TO 444
  504 D(12)=1
      DO 512 I=1,3
  512 D(I)=1
      GO TO 444
  506 D(13)=1
      DO 513 I=1,4
  513 D(I)=1
      GO TO 444
  507 D(14)=1
      DO 514 I=1,5
  514 D(I)=1
      GO TO 444
  509 D(15)=1
      DO 515 I=1,6
  515 D(I)=1
      GO TO 444
  510 D(8)=1
      DO 516 I=1,7
  516 D(I)=1
      GO TO 444
  511 PRINT 601
  601 FORMAT(5X,'ORDER EXCEEDS 10, VERIFY L VALUE',/)
```

```
  444 CALL REDUC1(T,X,XDOT,C,CD,CN,A,B,U,Z,D,OMEGA,L)
      ZIT=C(1)*1.+C(2)*T+C(3)*(SIN(OMEGA*T))
      ZEK=CD(7)*X(7)+CD(6)*X(6)+CD(5)*X(5)+CD(4)*X(4)
      ZEB=CD(3)*X(3)+CD(2)*X(2)+CD(1)*X(1)-ZIT
      XDOT(1)=X(2)
      XDOT(2)=X(3)
      XDOT(3)=D(1)*X(4)-D(9)*ZEB
      XDOT(4)=D(2)*X(5)-D(10)*(CD(4)*X(4)+ZEB)
      XDOT(5)=D(3)*X(6)-D(11)*(CD(5)*X(5)+CD(4)*X(4)+ZEB)
      XDOT(6)=D(4)*X(7)-D(12)*(CD(6)*X(6)+CD(5)*X(5)+CD(4)*X(4)+ZEB)
      XDOT(7)=D(5)*X(8)-D(13)*(ZEK+ZEB)
      XDOT(8)=D(6)*X(9)-D(14)*(CD(8)*X(8)+ZEK+ZEB)
      XDOT(9)=D(7)*X(10)-D(15)*(CD(9)*X(9)+CD(8)*X(8)+ZEK+ZEB)
      XDOT(10)=-D(8)*(CD(10)*X(10)+CD(9)*X(9)+CD(8)*X(8)+ZEK+ZEB)
      XDOT(11)=X(12)
      XDOT(12)=-A(2)*X(12)-Z(2)*X(11)+ZIT
      XDOT(13)=X(14)
      XDOT(14)=X(15)
      XDOT(15)=-Z(3)*X(15)-Z(4)*X(14)-Z(5)*X(13)+ZIT
      XDOT(16)=X(17)
      XDOT(17)=X(18)
      XDOT(18)=X(19)
      XDOT(19)=-Z(6)*X(19)-Z(7)*X(18)-Z(8)*X(17)-Z(9)*X(16)+ZIT
      X(20)=CN(10)*X(10)+CN(9)*X(9)+CN(8)*X(8)+CN(7)*X(7)+CN(6)*X(6)
     1+CN(5)*X(5)+CN(4)*X(4)+CN(3)*X(3)+CN(2)*X(2)+CN(1)*X(1)
      X(21)=B(2)*X(12)+U(1)*X(11)
      X(22)=U(2)*X(15)+U(3)*X(14)+U(4)*X(13)
      X(23)=U(5)*X(19)+U(6)*X(18)+U(7)*X(17)+U(8)*X(16)
      X(24)=X(20)-X(21)
      X(25)=X(20)-X(22)
      X(26)=X(20)-X(23)
      GO TO 444
      END
      SUBROUTINE REDUC1(/TC/,/XC/,/DX/,/C/,/CD/,/CN/,/A/,/B/,/U/,
     1/Z/,/D/,/OMEGA/,/L/)
      REAL*8 ITITLE(12),JTITLE(8),KTITLE(8),IBLANK/'        '/           INT12570
      DIMENSION X(30),BX(30),XC(30),C(15),IP(10),IG(10),PR(10),GR(10),  INT12580
     1TX(5),TY(5),X1(900),Y1(900),X2(900),Y2(900),X3(900),Y3(900),      INT12590
     2X4(900),Y4(900),CD(12),CN(12),A(4),B(4),U(8),Z(9),D(15)
      REAL LABEL,RUN(2)/' RUN',''    '/,BT(9)/' 1  ',' 2  ',' 3  ',' 4  'INT12610
     1,' 5  ',' 6  ',' 7  ',' 8  ',' 9  '/                              INT12620
      EQUIVALENCE (ITITLE(7),RUN(1))                                    INT12630
      INDIC = C(10)+0.0000001                                           INT12640
      GO TO (1, 2000, 50, 58, 88, 88),INDIC                             INT12650
C                                                                       INT12660
C         READ DATA AND PRINT RECORD.                                   INT12670
C                                                                       INT12680
    1 READ(5,100) (ITITLE(I), I=1,6)                                    INT12690
```

```
  100 FORMAT (10A8)                                                        INT12700
      READ (5,101)NR                                                       INT12710
  101 FORMAT (I1)                                                          INT12720
      NN=19
      NRC = 0                                                              INT12750
      GO TO 1000
 1000 NRC = NRC + 1                                                        INT12800
      WRITE (6,201) (ITITLE(I),I=1,6)                                      INT12810
  201 FORMAT (1H1,/////,36X,6A8)                                           INT12820
      IF(NRC.EQ.1.AND.NR.EQ.1) GO TO 5                                     INT12830
      WRITE(6,202)NR                                                       INT12840
  202 FORMAT (/,37X,I1,20H RUNS ARE CALLED FOR )                           INT12850
      GO TO 6                                                              INT12860
    5 WRITE(6,203)                                                         INT12870
  203 FORMAT (/,37X,21HONE RUN IS CALLED FOR ,///,18H INPUT DATA RECORD)INT12880
      GO TO 7                                                              INT12890
    6 WRITE(6,204)NRC                                                      INT12900
  204 FORMAT (///,34H INPUT DATA RECORD FOR RUN NUMBER   ,I1)              INT12910
    7 WRITE(6,205)NN                                                       INT12920
  205 FORMAT (//,22H ORDER OF EQUATIONS = ,I2)                             INT12930
      READ (5,103)TI,DT,TF1,DT2,TF2,DT3,TF3                                INT12940
  103 FORMAT (8F10.4)                                                      INT12950
      TF = TF1                                                             INT12960
      IF(DT2.NE.0.) GO TO 9                                                INT12970
      WRITE(6,206)TI,TF                                                    INT12980
  206 FORMAT (22H INITIAL TIME         = ,E10.4,/                          INT12990
     1        22H FINAL TIME           = ,E10.4)                           INT13000
      WRITE(6,207)DT                                                       INT13010
  207 FORMAT (22H STEP SIZE            = ,E10.4)                           INT13020
      GO TO 12                                                             INT13030
    9 IF(DT3.NE.0.) GO TO 11                                               INT13040
      TF = TF2                                                             INT13050
      WRITE(6,206) TI,TF                                                   INT13060
      WRITE(6,208)DT,TI,TF1,DT2,TF1,TF                                     INT13070
  208 FORMAT (22H STEP SIZE            = ,E10.4,13H BETWEEN T = ,E10.4,   INT13080
     1         9H AND T = ,E10.4)                                          INT13090
      GO TO 12                                                             INT13100
   11 TF=TF3                                                               INT13110
      WRITE(6,208) DT,TI,TF1,DT2,TF1,TF2,DT3,TF2,TF                        INT13120
   12 READ(5,103) (C(I),I=1,8)                                             INT13130
      READ (5,103)(X(I),I=1,NN)                                            INT13140
      J = 0                                                                INT13150
      DO 14 I=1,8                                                          INT13160
      IF(C(I).NE.0.) J=J+1                                                 INT13170
   14 CONTINUE                                                             INT13180
      K = 0                                                                INT13190
      DO 16 I=1,NN                                                         INT13200
      IF(X(I).NE.0.) K=K+1                                                 INT13210
```

```
   16 CONTINUE                                                          INT13220
      IF(J - 1)17,18,19                                                 INT13230
   17 WRITE (6,209)                                                     INT13240
  209 FORMAT (/,34H ALL THE CONSTANTS, C(I), ARE ZERO )                 INT13250
      GO TO 423                                                         INT13260
   18 WRITE (6,210)                                                     INT13270
  210 FORMAT (/,30H THE ONLY NON-ZERO CONSTANT IS )                     INT13280
      GO TO 420                                                         INT13290
   19 WRITE (6,211)                                                     INT13300
  211 FORMAT (/,35H THE NON-ZERO CONSTANTS, C(I), ARE )                 INT13310
  420 DO 422 I=1,8                                                      INT13320
      IF(C(I).NE.0.) WRITE(6,212) I,C(I)                                INT13330
  212 FORMAT (14X,2HC(,I2,4H) = ,E10.4)                                 INT13340
  422 CONTINUE                                                          INT13350
  423 IF(K - 1)424,425,426                                              INT13360
  424 WRITE (6,1209)                                                    INT13370
 1209 FORMAT (/,36H ALL THE INITIAL CONDITIONS ARE ZERO )               INT13380
      GO TO 20                                                          INT13390
  425 WRITE (6,1210)                                                    INT13400
 1210 FORMAT (/,39H THE ONLY NON-ZERO INITIAL CONDITION IS )            INT13410
      GO TO 427                                                         INT13420
  426 WRITE (6,1211)                                                    INT13430
 1211 FORMAT (/,36H THE NON-ZERO INITIAL CONDITIONS ARE )               INT13440
  427 DO 429 I=1,NN                                                     INT13450
      IF(X(I).NE.0.) WRITE(6,1212) I,X(I)                               INT13460
 1212 FORMAT (14X,2HX(,I2,4H) = ,E10.4)                                 INT13470
  429 CONTINUE                                                          INT13480
   20 READ (5,104) (JTITLE(I),IP(I),I=1,8)                              INT13490
  104 FORMAT(8(A8,I2))                                                  INT13500
C                                                                       INT13510
C     CHECK FOR THE NUMBER OF COLUMNS CALLED FOR BY LOCATING FIRST      INT13520
C     BLANK COLUMN HEADING                                              INT13530
C                                                                       INT13540
      DO 21 J=1,8                                                       INT13550
      IF(JTITLE(J).EQ.IBLANK) GO TO 22                                  INT13560
   21 CONTINUE                                                          INT13570
      J = 9                                                             INT13580
   22 JJ = J - 1                                                        INT13590
C                                                                       INT13600
C     JJ IS NOW THE NUMBER OF COLUMNS. REPEAT WITH THE GRAPHS.          INT13610
C                                                                       INT13620
      READ (5,105)(KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,7,2)         INT13630
  105 FORMAT (4(2A8,2I2))                                               INT13640
      DO 24 K=1,7,2                                                     INT13650
      IF(KTITLE(K).EQ.IBLANK.AND.KTITLE(K+1).EQ.IBLANK) GO TO 25        INT13660
   24 CONTINUE                                                          INT13670
      K = 8                                                             INT13680
   25 KK = K/2                                                          INT13690
```

```
      KKK = KK*2                                                        INT13700
      MULTIP = 0                                                        INT13710
      IF(KK.NE.1) GO TO 306                                             INT13720
      IF(IG(3) + IG(4).EQ.0) GO TO 306                                  INT13730
      IF(IG(5) + IG(6).NE.0) GO TO 303                                  INT13740
      MULTIP = 2                                                        INT13750
      KKK = 4                                                           INT13760
      GO TO 306                                                         INT13770
  303 IF(IG(7) + IG(8).NE.0) GO TO 305                                  INT13780
      MULTIP = 3                                                        INT13790
      KKK = 6                                                           INT13800
      GO TO 306                                                         INT13810
  305 MULTIP = 4                                                        INT13820
      KKK = 8                                                           INT13830
C                                                                       INT13840
C     IF MULTIP = 0, KK IS THE NUMBER OF SINGLE CURVE GRAPHS. OTHERWISE INT13850
C     MULTIP IS THE NUMBER OF CURVES ON A SINGLE GRAPH.                 INT13860
C                                                                       INT13870
  306 IF(JJ.EQ.0) GO TO 27                                              INT13880
      WRITE(6,214) (JTITLE(I),IP(I),I=1,JJ)                             INT13890
  214 FORMAT (///,56H THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLEINT13900
     1S ARE ,//,(10X,A8,4X,2HX(,I2,1H)))                                INT13910
      GO TO 28                                                          INT13920
   27 WRITE(6,215)                                                      INT13930
  215 FORMAT (///,25H NO PRINTOUT IS REQUIRED   )                       INT13940
   28 IF(KK.EQ.0) GO TO 308                                             INT13950
      IF(MULTIP.NE.0) GO TO 309                                         INT13960
      IF(KK.NE.1) GO TO 307                                             INT13970
      WRITE (6,216) KTITLE(1),KTITLE(2),IG(1),IG(2)                     INT13980
  216 FORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARINT13990
     1E ,//,10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H))                       INT14000
      GO TO 31                                                          INT14010
  307 WRITE (6,217) (KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,KKK,2)     INT14020
  217 FORMAT (///,64H THE INDIVIDUAL GRAPH TITLES AND THE CORRESPONCING INT14030
     1VARIABLES ARE ,//,(10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H)))         INT14040
      GO TO 31                                                          INT14050
  308 WRITE (6,1217)                                                    INT14060
 1217 FORMAT(///,24H NO GRAPHS ARE REQUIRED   )                         INT14070
      GO TO 31                                                          INT14080
  309 WRITE(6,1220)                                                     INT14090
 1220 FORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARINT14100
     1E ,/)                                                             INT14110
      WRITE(6,1221) KTITLE(1),KTITLE(2),(IG(I),IG(I+1),I=1,KKK,2)       INT14120
 1221 FORMAT (10X,2A8,4X,2HX(,I2,8H) VS. X(,I2,1H),/, (30X,2HX(,I2,     INT14130
     1          8H) VS. X(,I2,1H)))                                     INT14140
C                                                                       INT14150
C     THIS ENDS THE BOOK-KEEPING. INITIALIZE BEFORE ENTERING MAIN LOOP. INT14160
C                                                                       INT14170
```

```
   31 IPAGE = 0                                                          INT14180
      T = TI                                                             INT14190
      NOPTS = 0                                                          INT14200
      NUMPTS = 0                                                         INT14210
      ITITLE(8)=IBLANK                                                   INT14220
      ITITLE(11)=IBLANK                                                  INT14230
      ITITLE(12)=IBLANK                                                  INT14240
      RUN(2)=BT(NRC)                                                     INT14250
      C(11) = 20.                                                        INT14260
      C(12) = 5.                                                         INT14270
      C(13) = DT                                                         INT14280
      DO 42 I=1,NN                                                       INT14290
   42 XC(I) = X(I)                                                       INT14300
      TC = T                                                             INT14310
      C(10) = 2.                                                         INT14320
      RETURN                                                             INT14330
C                                                                        INT14340
 2000 IF(JJ.EQ.0) GO TO 54                                               INT14350
C                                                                        INT14360
      INCPR = C(11)+0.0000001                                            INT14370
      C(11) = 20.                                                        INT14380
      IF( MOD (NOPTS,50*INCPR).EQ.0) GO TO 46                            INT14390
      IF( MOD (NOPTS,10*INCPR).EQ.0) GO TO 47                            INT14400
      IF( MOD (NOPTS,    INCPR))54,48,54                                 INT14410
   46 IPAGE = IPAGE + 1                                                  INT14420
      IF(NR.EQ.1) GO TO 1047                                             INT14430
      WRITE (6,218) (ITITLE(I),I=1,6),IPAGE,ITITLE(7),(JTITLE(I),I=1,8)  INT14440
      WRITE(6,219)                                                       INT14450
      GO TO 47                                                           INT14460
 1047 WRITE (6,1218)(ITITLE(I),I=1,6),IPAGE,(JTITLE(I),I=1,8)            INT14470
      WRITE (6,219)                                                      INT14480
   47 WRITE(6,219)                                                       INT14490
  218 FORMAT (1H1,///,20X,6A8,10X,5HPAGE ,I1,14H OF OUTPUT FOR,A8,/////,INT14500
     1        11X,8(A8,5X))                                              INT14510
 1218 FORMAT (1H1,///,20X,6A8,30X,5HPAGE ,I1,/////,11X,8(A8,5X))         INT14520
  219 FORMAT (1H  )                                                      INT14530
   48 DO 49 I=1,NN                                                       INT14540
   49 XC(I) = X(I)                                                       INT14550
      TC = T                                                             INT14560
      C(10) = 3.                                                         INT14570
      RETURN                                                             INT14580
C                                                                        INT14590
   50 DO 53 I=1,JJ                                                       INT14600
C                                                                        INT14610
      PR(I) = T                                                          INT14620
      IF(IP(I).NE.0) PR(I)=XC(IP(I))                                     INT14630
   53 CONTINUE                                                           INT14640
      WRITE (6,220)(PR(I),I=1,JJ)                                        INT14650
```

```
  220 FORMAT (7X, 8E13.5)                                              INT14660
   54 IF(KK.EQ.0) GO TO 62                                             INT14670
      INCGR = C(12)+0.0000001                                          INT14680
      C(12) = 5.                                                       INT14690
      IF( MOD (NOPTS, INCGR).NE.0) GO TO 62                            INT14700
      DO 57 I=1,NN                                                     INT14710
   57 XC(I) = X(I)                                                     INT14720
      TC = T                                                           INT14730
      C(10) = 4.                                                       INT14740
      RETURN                                                           INT14750
C                                                                      INT14760
   58 DO 61 I=1,KKK                                                    INT14770
C                                                                      INT14780
      GR(I) = T                                                        INT14790
      IF(IG(I).NE.0) GR(I)=XC(IG(I))                                   INT14800
   61 CONTINUE                                                         INT14810
      IF (KKK .GE. 8) GO TO 1610                                       INT14820
      KP1 = KKK + 1                                                    INT14830
      DO 1612 I=KP1,8                                                  INT14840
 1612 GR(I) = 0.                                                       INT14850
 1610 NUMPTS = NUMPTS + 1                                              INT14860
      Y1(NUMPTS) = GR(1)                                               INT14870
      X1(NUMPTS) = GR(2)                                               INT14880
      Y2(NUMPTS) = GR(3)                                               INT14890
      X2(NUMPTS) = GR(4)                                               INT14900
      Y3(NUMPTS) = GR(5)                                               INT14910
      X3(NUMPTS) = GR(6)                                               INT14920
      Y4(NUMPTS) = GR(7)                                               INT14930
      X4(NUMPTS) = GR(8)                                               INT14940
   62 NOPTS = NOPTS + 1                                                INT14950
      IF(NUMPTS.LT.900) GO TO 64                                       INT14960
      WRITE (6,221)                                                    INT14970
  221 FORMAT (//////,25H STOP AT 900 GRAPH POINTS    )                 INT14980
      GO TO 91                                                         INT14990
   64 IF(NOPTS.LT.4500) GO TO 66                                       INT15000
      WRITE (6,222)                                                    INT15010
  222 FORMAT (//////,31H STOP AT 4500 INTEGRATION STEPS   )            INT15020
      GO TO 91                                                         INT15030
   66 IF(IPAGE - 9)69,67,68                                            INT15040
   67 IF( MOD (NOPTS , 50*INCPR).NE.0) GO TO 69                        INT15050
   68 WRITE (6,223)                                                    INT15060
  223 FORMAT (//////,27H STOP AT 9 PAGES OF OUTPUT  )                  INT15070
      GO TO 91                                                         INT15080
   69 DO 70 I=1,NN                                                     INT15090
      IF(ABS(X(I)).GT.1.E+12) GO TO 71                                 INT15100
   70 CONTINUE                                                         INT15110
      GO TO 72                                                         INT15120
   71 WRITE (6,224)                                                    INT15130
```

```
  224 FORMAT (//////,76H STOP WITH THE ABSOLUTE VALUE OF A DEPENDENT VARINT15140
     1IABLE GREATER THAN 1.0E+12.   ,/,57H INTEGRATION PROBABLY UNSTABLE.INT15150
     2 TRY A SMALLER STEP SIZE. ,26HNO GRAPHS WILL BE PLOTTED.        )  INT15160
      GO TO 330                                                         INT15170
   72 CT = C(13)                                                        INT15180
      IF(TI.GT.TF) GO TO 80                                             INT15190
   73 IF(T.LT.TF) GO TO 75                                              INT15200
   74 WRITE (6,225)                                                     INT15210
  225 FORMAT (//////,26H NORMAL STOP AT FINAL TIME    )                 INT15220
      GO TO 91                                                          INT15230
   75 IF(T.GE.TF1) GO TO 77                                             INT15240
   76 C(13) = DT                                                        INT15250
      GO TO 87                                                          INT15260
   77 IF(T.GE.TF2) GO TO 79                                             INT15270
   78 C(13) = DT2                                                       INT15280
      GO TO 87                                                          INT15290
   79 C(13) = DT3                                                       INT15300
      GO TO 87                                                          INT15310
   80 IF(TF.GE.T) GO TO 74                                              INT15320
      IF(TF1.LT.T) GO TO 76                                             INT15330
      IF(TF2 - T)78,79,79                                               INT15340
   87 C(10) = 5.                                                        INT15350
C                                                                       INT15360
   88 CALL RKUTTA (NN,T,X,DT,C,TC,XC,DX)                                INT15370
C                                                                       INT15380
      IF(C(10).EQ.6.) RETURN                                            INT15390
   90 T = T + DT                                                        INT15400
      GO TO 2000                                                        INT15410
   91 IF(KK.EQ.0) GO TO 330                                             INT15420
      IF(MULTIP.NE.0) GO TO 97                                          INT15430
C                                                                       INT15440
C     PRINT PLOT UP TO 4 INDIVIDUAL CURVES                              INT15450
C                                                                       INT15460
      NUMPTS=-NUMPTS                                                    INT15470
      DO 310 II=1,KK                                                    INT15480
      WRITE(6,9998)                                                     INT15490
 9998 FORMAT(1H1)                                                       INT15500
      ITITLE(9)=KTITLE(2*II-1)                                          INT15510
      ITITLE(10)=KTITLE(2*II)                                           INT15520
      GO TO (311,312,313,314),II                                        INT15530
  311 CALL PLOTP(X1,Y1,NUMPTS,0)                                        INT15540
      GO TO 310                                                         INT15550
  312 CALL PLOTP(X2,Y2,NUMPTS,0)                                        INT15560
      GO TO 310                                                         INT15570
  313 CALL PLOTP(X3,Y3,NUMPTS,0)                                        INT15580
      GO TO 310                                                         INT15590
  314 CALL PLOTP(X4,Y4,NUMPTS,0)                                        INT15600
  310 WRITE(6,9999) ITITLE                                              INT15610
```

```
 9999 FORMAT(1H0,8X,12A8)
      GO TO 330
C
C     PLOT DUMMY CURVE ALONG AXES TO SET SCALES FOR MULTIPLE PLOT
C
   97 BIGX = 0.
      BIGY = 0.
      SMLX = 0.
      SMLY = 0.
      DO 1970 I=1,NUMPTS
      XMAX =AMAX1  ( X1(I), X2(I), X3(I), X4(I))
      YMAX =AMAX1  ( Y1(I), Y2(I), Y3(I), Y4(I))
      XMIN =AMIN1  (X1(I), X2(I), X3(I), X4(I))
      YMIN =AMIN1 ( Y1(I), Y2(I), Y3(I), Y4(I))
      IF(BIGX.LT.XMAX) BIGX=XMAX
      IF(BIGY.LT.YMAX) BIGY=YMAX
      IF(SMLX.GT.XMIN) SMLX=XMIN
      IF(SMLY.GT.YMIN) SMLY=YMIN
 1970 CONTINUE
      TX(1) = 0.
      TX(2) = 0.
      TX(3) = 0.
      TX(4) = SMLX
      TX(5) = BIGX
      TY(1) = BIGY
      TY(2) = SMLY
      TY(3) = 0.
      TY(4) = 0.
      TY(5) = 0.
      WRITE(6,9998)
      ITITLE(9) = KTITLE(1)
      ITITLE(10)= KTITLE(2)
      NT=-5
      CALL PLOTP(TX,TY,NT,1)
      MODCUR  = 2
      DO 410 II=1,MULTIP
      IF(II.EQ.MULTIP) MODCUR=3
      GO TO  (411,412,413,414),II
  411 CALL PLOTP(X1,Y1,NUMPTS,MODCUR)
      GO TO 410
  412 CALL PLOTP(X2,Y2,NUMPTS,MODCUR)
      GO TO 410
  413 CALL PLOTP(X3,Y3,NUMPTS,MODCUR)
      GO TO 410
  414 CALL PLOTP(X4,Y4,NUMPTS,MODCUR)
  410 CONTINUE
      WRITE(6,9999) ITITLE
C
```

```
330 IF(NRC.NE.NR) GO TO 1000
    IF(NR.GT.1) GO TO 333
    WRITE(6,226)
226 FORMAT(//,43H THE ONE RUN CALLED FOR HAS BEEN COMPLETED. ,//)
    STOP
333 WRITE(6,227)NR
227 FORMAT(//,5H THE ,I1,37H RUNS CALLED FOR HAVE BEEN COMPLETED.,//)
    STOP
    END
    SUBROUTINE RKUTTA(/NN/,/T/,/X/,/DT/,/C/,/TC/,/XC/,/DX/)
    DIMENSION X(30), C(15), XC(30), DX(30), CT(4), AK(4,30)
    REAL*8 AK,CT
    INDIC = C(10) - 4.0+0.0000001
    IF(INDIC.GT.1) GO TO 3
    CT(1) = 0.0D0
    CT(2) = 0.5D0
    CT(3) = 0.5D0
    CT(4) = 1.0D0
    II=0
  7 II=II+1
    TC = T + CT(II)*DT
    DO 2 J=1,NN
  2 XC(J) = X(J) + CT(II)*AK(II-1, J)
    C(10) = 6.0
    RETURN
  3 DO 4 J=1,NN
  4 AK(II, J) = DT*DX(J)
    IF(II.LT.4) GO TO 7
    DO 5 J=1,NN
  5 X(J)=X(J)+(AK(1,J)+2.0*(AK(2,J)+AK(3,J))+AK(4,J))/6.0
    C(10) = 7.0
    RETURN
    END
C       SUBROUTINE PRQD
C
C       PURPOSE
C          CALCULATE ALL REAL AND COMPLEX ROOTS OF A GIVEN POLYNOMIAL
C          WITH REAL COEFFICIENTS.
C
    SUBROUTINE PRQD(C,IC,Q,E,POL,IR,IER)
C
C    DIMENSIONED DUMMY VARIABLES
    DIMENSION E(1),Q(1),C(1),POL(1)
C
C       NORMALIZATION OF GIVEN POLYNOMIAL
C          TEST OF DIMENSION
C       IR CONTAINS INDEX OF HIGHEST COEFFICIENT
    IER=0
```

```
      IR=IC                                                             PRQD 710
      EPS=1.E-6                                                         PRQD 720
      TOL=1.E-3                                                         PRQD 730
      LIMIT=10*IC                                                       PRQD 740
      KOUNT=0                                                           PRQD 750
    1 IF(IR-1)79,79,2                                                   PRQD 760
C                                                                       PRQD 770
C         CROP TRAILING ZERO COEFFICIENTS                               PRQD 780
    2 IF(C(IR))4,3,4                                                    PRQD 790
    3 IR=IR-1                                                           PRQD 800
      GOTO 1                                                            PRQD 810
C                                                                       PRQD 820
C             REARRANGEMENT OF GIVEN POLYNOMIAL                         PRQD 830
C         EXTRACTION OF ZERO ROOTS                                      PRQD 840
    4 Q=1./C(IR)                                                        PRQD 850
      IEND=IR-1                                                         PRQD 860
      ISTA=1                                                            PRQD 870
      NSAV=IR+1                                                         PRQD 880
      JBEG=1                                                            PRQD 890
C                                                                       PRQD 900
C         Q(J)=1.                                                       PRQD 910
C         Q(J+I)=C(IR-I)/C(IR)                                          PRQD 920
C         C(IR)=C(J)/C(IR)                                              PRQD 930
C         WHERE J IS THE INDEX OF THE LOWEST NONZERO COEFFICIENT        PRQD 940
      DO 9 I=1,IR                                                       PRQD 950
      J=NSAV-I                                                          PRQD 960
      IF(C(I))7,5,7                                                     PRQD 970
    5 GOTO(6,8),JBEG                                                    PRQD 980
    6 NSAV=NSAV+1                                                       PRQD 990
      Q(ISTA)=0.                                                        PRQD1000
      E(ISTA)=0.                                                        PRQD1010
      ISTA=ISTA+1                                                       PRQD1020
      GOTO 9                                                            PRQD1030
    7 JBEG=2                                                            PRQD1040
    8 Q(J)=C(I)*Q                                                       PRQD1050
      C(I)=Q(J)                                                         PRQD1060
    9 CONTINUE                                                          PRQD1070
C                                                                       PRQD1080
C             INITIALIZATION                                            PRQD1090
      ESAV=0.                                                           PRQD1100
      Q(ISTA)=0.                                                        PRQD1110
   10 NSAV=IR                                                           PRQD1120
C                                                                       PRQD1130
C         COMPUTATION OF DERIVATIVE                                     PRQD1140
      EXPT=IR-ISTA                                                      PRQD1150
      E(ISTA)=EXPT                                                      PRQD1160
      DO 11 I=ISTA,IEND                                                 PRQD1170
      EXPT=EXPT-1.0                                                     PRQD1180
```

```
      POL(I+1)=EPS*ABS(Q(I+1))+EPS                                      PRQD1190
   11 E(I+1)=Q(I+1)*EXPT                                                PRQD1200
C                                                                       PRQD1210
C        TEST OF REMAINING DIMENSION                                    PRQD1220
      IF(ISTA-IEND)12,20,60                                             PRQD1230
   12 JEND=IEND-1                                                       PRQD1240
C                                                                       PRQD1250
C        COMPUTATION OF S-FRACTION                                      PRQD1260
      DO 19 I=ISTA,JEND                                                 PRQD1270
      IF(I-ISTA)13,16,13                                                PRQD1280
   13 IF(ABS(E(I))-POL(I+1))14,14,16                                    PRQD1290
C                                                                       PRQD1300
C        THE GIVEN POLYNOMIAL HAS MULTIPLE ROOTS, THE COEFFICIENTS OF   PRQD1310
C        THE COMMON FACTOR ARE STORED FROM Q(NSAV) UP TO Q(IR)          PRQD1320
   14 NSAV=I                                                            PRQD1330
      DO 15 K=I,JEND                                                    PRQD1340
      IF(ABS(E(K))-POL(K+1))15,15,80                                    PRQD1350
   15 CONTINUE                                                          PRQD1360
      GOTO 21                                                           PRQD1370
C                                                                       PRQD1380
C           EUCLIDEAN ALGORITHM                                         PRQD1390
   16 DO 19 K=I,IEND                                                    PRQD1400
      E(K+1)=E(K+1)/E(I)                                                PRQD1410
      Q(K+1)=E(K+1)-Q(K+1)                                              PRQD1420
      IF(K-I)18,17,18                                                   PRQD1430
C                                                                       PRQD1440
C        TEST FOR SMALL DIVISOR                                         PRQD1450
   17 IF(ABS(Q(I+1))-POL(I+1))80,80,19                                  PRQD1460
   18 Q(K+1)=Q(K+1)/Q(I+1)                                              PRQD1470
      POL(K+1)=POL(K+1)/ABS(Q(I+1))                                     PRQD1480
      E(K)=Q(K+1)-E(K)                                                  PRQD1490
   19 CONTINUE                                                          PRQD1500
   20 Q(IR)=-Q(IR)                                                      PRQD1510
C                                                                       PRQD1520
C           THE DISPLACEMENT EXPT IS SET TO 0 AUTOMATICALLY.            PRQD1530
C           E(ISTA)=0.,Q(ISTA+1),...,E(NSAV-1),Q(NSAV),E(NSAV)=0.,      PRQD1540
C           FORM A DIAGONAL OF THE QD-ARRAY.                            PRQD1550
C        INITIALIZATION OF BOUNDARY VALUES                              PRQD1560
   21 E(ISTA)=0.                                                        PRQD1570
      NRAN=NSAV-1                                                       PRQD1580
   22 E(NRAN+1)=0.                                                      PRQD1590
C                                                                       PRQD1600
C           TEST FOR LINEAR OR CONSTANT FACTOR                          PRQD1610
C        NRAN-ISTA IS DEGREE-1                                          PRQD1620
      IF(NRAN-ISTA)24,23,31                                             PRQD1630
C                                                                       PRQD1640
C        LINEAR FACTOR                                                  PRQD1650
   23 Q(ISTA+1)=Q(ISTA+1)+EXPT                                          PRQD1660
```

```
      E(ISTA+1)=0.
C
C        TEST FOR UNFACTORED COMMON DIVISOR
   24 E(ISTA)=ESAV
      IF(IR-NSAV)60,60,25
C
C        INITIALIZE QD-ALGORITHM FOR COMMON DIVISOR
   25 ISTA=NSAV
      ESAV=E(ISTA)
      GOTO 10
C
C        COMPUTATION OF ROOT PAIR
   26 P=P+EXPT
C
C          TEST FOR REALITY
      IF(C)27,28,28
C
C          COMPLEX ROOT PAIR
   27 Q(NRAN)=P
      Q(NRAN+1)=P
      E(NRAN)=T
      E(NRAN+1)=-T
      GOTO 29
C
C          REAL ROOT PAIR
   28 Q(NRAN)=P-T
      Q(NRAN+1)=P+T
      E(NRAN)=0.
C
C          REDUCTION OF DEGREE BY 2 (DEFLATION)
   29 NRAN=NRAN-2
      GOTO 22
C
C        COMPUTATION OF REAL ROOT
   30 Q(NRAN+1)=EXPT+P
C
C          REDUCTION OF DEGREE BY 1 (DEFLATION)
      NRAN=NRAN-1
      GOTO 22
C
C        START QD-ITERATION
   31 JBEG=ISTA+1
      JEND=NRAN-1
      TEPS=EPS
      TDELT=1.E-2
   32 KCUNT=KCUNT+1
      P=Q(NRAN+1)
      R=ABS(E(NRAN))
```

```
C                                                                       PRQD2150
C             TEST FOR CONVERGENCE                                      PRQD2160
      IF(R-TEPS)30,30,33                                                PRQD2170
   33 S=ABS(E(JEND))                                                    PRQD2180
C                                                                       PRQD2190
C         IS THERE A REAL ROOT NEXT                                     PRQD2200
      IF(S-R)38,38,34                                                   PRQD2210
C                                                                       PRQD2220
C         IS DISPLACEMENT SMALL ENOUGH                                  PRQD2230
   34 IF(R-TDELT)36,35,35                                               PRQD2240
   35 P=0.                                                              PRQD2250
   36 O=P                                                               PRQD2260
      DO 37 J=JBEG,NRAN                                                 PRQD2270
      Q(J)=Q(J)+E(J)-E(J-1)-O                                           PRQD2280
C                                                                       PRQD2290
C             TEST FOR SMALL DIVISOR                                    PRQD2300
      IF(ABS(Q(J))-POL(J))81,81,37                                      PRQD2310
   37 E(J)=Q(J+1)*E(J)/Q(J)                                             PRQD2320
      Q(NRAN+1)=-E(NRAN)+Q(NRAN+1)-O                                    PRQD2330
      GOTO 54                                                           PRQD2340
C                                                                       PRQD2350
C         CALCULATE DISPLACEMENT FOR DOUBLE ROOTS                       PRQD2360
C             QUADRATIC EQUATION FOR DOUBLE ROOTS                       PRQD2370
C             X**2-(Q(NRAN)+Q(NRAN+1)+E(NRAN))*X+Q(NRAN)*Q(NRAN+1)=0    PRQD2380
   38 P=0.5*(Q(NRAN)+E(NRAN)+Q(NRAN+1))                                 PRQD2390
      O=P*P-Q(NRAN)*Q(NRAN+1)                                           PRQD2400
      T=SQRT(ABS(O))                                                    PRQD2410
C                                                                       PRQD2420
C         TEST FOR CONVERGENCE                                          PRQD2430
      IF(S-TEPS)26,26,39                                                PRQD2440
C                                                                       PRQD2450
C         ARE THERE COMPLEX ROOTS                                       PRQD2460
   39 IF(O)43,40,40                                                     PRQD2470
   40 IF(P)42,41,41                                                     PRQD2480
   41 T=-T                                                              PRQD2490
   42 P=P+T                                                             PRQD2500
      R=S                                                               PRQD2510
      GOTO 34                                                           PRQD2520
C                                                                       PRQD2530
C         MODIFICATION FOR COMPLEX ROOTS                                PRQD2540
C         IS DISPLACEMENT SMALL ENOUGH                                  PRQD2550
   43 IF(S-TDELT)44,35,35                                               PRQD2560
C                                                                       PRQD2570
C         INITIALIZATION                                                PRQD2580
   44 O=Q(JBEG)+E(JBEG)-P                                               PRQD2590
C                                                                       PRQD2600
C         TEST FOR SMALL DIVISOR                                        PRQD2610
      IF(ABS(O)-POL(JBEG))81,81,45                                      PRQD2620
```

```
   45 T=(T/O)**2                                                          PRQD2630
      U=E(JBEG)*Q(JBEG+1)/(O*(1.+T))                                      PRQD2640
      V=O+U                                                               PRQD2650
      KOUNT=KOUNT+2                                                       PRQD2660
C                                                                         PRQD2670
C        THREEFOLD LOOP FOR COMPLEX DISPLACEMENT                          PRQD2680
      DO 53 J=JBEG,NRAN                                                   PRQD2690
      O=Q(J+1)+E(J+1)-U-P                                                 PRQD2700
C                                                                         PRQD2710
C        TEST FOR SMALL DIVISOR                                           PRQD2720
      IF(ABS(V)-POL(J))46,46,49                                           PRQD2730
   46 IF(J-NRAN)81,47,81                                                  PRQD2740
   47 EXPT=EXPT+P                                                         PRQD2750
      IF(ABS(E(JEND))-TOL)48,48,81                                        PRQD2760
   48 P=0.5*(V+O-E(JEND))                                                 PRQD2770
      O=P*P-(V-U)*(O-U*T-O*W*(1.+T)/Q(JEND))                              PRQD2780
      T=SQRT(ABS(O))                                                      PRQD2790
      GOTO 26                                                             PRQD2800
C                                                                         PRQD2810
C           TEST FOR SMALL DIVISOR                                        PRQD2820
   49 IF(ABS(O)-POL(J+1))46,46,50                                         PRQD2830
   50 W=U*O/V                                                             PRQD2840
      T=T*(V/O)**2                                                        PRQD2850
      Q(J)=V+W-E(J-1)                                                     PRQD2860
      U=0.                                                                PRQD2870
      IF(J-NRAN)51,52,52                                                  PRQD2880
   51 U=Q(J+2)*E(J+1)/(O*(1.+T))                                          PRQD2890
   52 V=O+U-W                                                             PRQD2900
C                                                                         PRQD2910
C        TEST FOR SMALL DIVISOR                                           PRQD2920
      IF(ABS(Q(J))-POL(J))81,81,53                                        PRQD2930
   53 E(J)=W*V*(1.+T)/Q(J)                                                PRQD2940
      Q(NRAN+1)=V-E(NRAN)                                                 PRQD2950
   54 EXPT=EXPT+P                                                         PRQD2960
      TEPS=TEPS*1.1                                                       PRQD2970
      TDELT=TDELT*1.1                                                     PRQD2980
      IF(KOUNT-LIMIT)32,55,55                                             PRQD2990
C                                                                         PRQD3000
C        NO CONVERGENCE WITH FEASIBLE TOLERANCE                           PRQD3010
C           ERROR RETURN IN CASE OF UNSATISFACTORY CONVERGENCE            PRQD3020
   55 IER=1                                                               PRQD3030
C                                                                         PRQD3040
C        REARRANGE CALCULATED ROOTS                                       PRQD3050
   56 IEND=NSAV-NRAN-1                                                    PRQD3060
      E(ISTA)=ESAV                                                        PRQD3070
      IF(IEND)59,59,57                                                    PRQD3080
   57 DO 58 I=1,IEND                                                      PRQD3090
      J=ISTA+I                                                            PRQD3100
```

```
      K=NRAN+1+I
      E(J)=E(K)
   58 Q(J)=Q(K)
   59 IR=ISTA+IEND
C
C        NORMAL RETURN
   60 IR=IR-1
      IF(IR)78,78,61
C
C        REARRANGE CALCULATED ROOTS
   61 DO 62 I=1,IR
      Q(I)=Q(I+1)
   62 E(I)=E(I+1)
C
C        CALCULATE COEFFICIENT VECTOR FROM ROOTS
      POL(IR+1)=1.
      IEND=IR-1
      JBEG=1
      DO 69 J=1,IR
      ISTA=IR+1-J
      O=0.
      P=Q(ISTA)
      T=E(ISTA)
      IF(T)65,63,65
C
C        MULTIPLY WITH LINEAR FACTOR
   63 DO 64 I=ISTA,IR
      POL(I)=O-P*POL(I+1)
   64 O=POL(I+1)
      GOTO 69
   65 GOTO(66,67),JBEG
   66 JBEG=2
      POL(ISTA)=0.
      GOTO 69
C
C        MULTIPLY WITH QUADRATIC FACTOR
   67 JBEG=1
      U=P*P+T*T
      P=P+P
      DO 68 I=ISTA,IEND
      POL(I)=O-P*POL(I+1)+U*POL(I+2)
   68 O=POL(I+1)
      POL(IR)=O-P
   69 CONTINUE
      IF(IER)78,70,78
C
C        COMPARISON OF COEFFICIENT VECTORS, IE. TEST OF ACCURACY
   70 P=0.
```

```
      DO 75 I=1,IR                                                      PRQD3590
      IF(C(I))72,71,72                                                  PRQD3600
   71 Q=ABS(POL(I))                                                     PRQD3610
      GOTO 73                                                           PRQD3620
   72 Q=ABS((POL(I)-C(I))/C(I))                                         PRQD3630
   73 IF(P-Q)74,75,75                                                   PRQD3640
   74 P=Q                                                               PRQD3650
   75 CONTINUE                                                          PRQD3660
      IF(P-TOL)77,76,76                                                 PRQD3670
   76 IER=-1                                                            PRQD3680
   77 Q(IR+1)=P                                                         PRQD3690
      E(IR+1)=0.                                                        PRQD3700
   78 RETURN                                                            PRQD3710
C                                                                       PRQD3720
C        ERROR RETURNS                                                  PRQD3730
C           ERROR RETURN FOR POLYNOMIALS OF DEGREE LESS THAN 1          PRQD3740
   79 IER=2                                                             PRQD3750
      IR=0                                                              PRQD3760
      RETURN                                                            PRQD3770
C                                                                       PRQD3780
C           ERROR RETURN IF THERE EXISTS NO S-FRACTION                  PRQD3790
   80 IER=4                                                             PRQD3800
      IR=ISTA                                                           PRQD3810
      GOTO 60                                                           PRQD3820
C                                                                       PRQD3830
C           ERROR RETURN IN CASE OF INSTABLE QD-ALGORITHM               PRQD3840
   81 IER=3                                                             PRQD3850
      GOTO 56                                                           PRQD3860
      END                                                               PRQD3870
```

BIBLIOGRAPHY

1. Hutton, M. F. and Friedland, B., "Routh Approximations of Reducing Order of Linear, Time-Invariant Systems," IEEE Transactions on Automatic Control, Vol. AC-20, pp 329-337.

2. Cantalapiedra, F., Low-Order Models for Dynamic Systems, MS Thesis, N. P. G. School, Monterey, California, June, 1972.

3. "Subroutine PRQD," W. R. Church Computer Facicliy Subroutine Library, N. P. G. School, Monterey, California.

4. Ward, J. R., "Subroutine INTEG1," W. R. Church Computer Facility Subroutine Library, N. P. G. School, Monterey, California.

5. Davison, E. J., "A Method for Simplifying Linear Dynamic Systems," IEEE Transactions on Automatic Control, Vol. AC-11, 1966.

6. Lal, M. and Van Valkenburg, M. E., "A Model-Reduction Method for Large Linear Systems," Ninth Annual Asilomar Conference on Circuits, Systems, and Computers, Nov, 1975.

7. Computer and Electrical Engineering Report, Vol. 2, pp 117-123, Computer Determination of Low-Order Models for High Order Systems, by G. J. Thaler, 22 May, 1974.

INITIAL DISTRIBUTION LIST

		Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Professor A. Gerba, Jr. Code 52 Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
5.	Professor G. J. Thaler, Code 52 Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6.	LT Jerry D. Thompson 62 West South Street Mooresville, Indiana 46158	1

Thesis
T43525 Thompson 165854
c.1 Reduced order approximations to higher order linear systems.